

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

CS409 Handouts

Introduction to Database Administration

Made by Mah jabeen
0321 2711298
[mahjabeen97869@gmail.com]

TOPIC 01

Defining Data, Information & Database

Defining Data, Information & Database

Data, Information From Data to Information

For Knowledge

For Business Intelligence

Defining Data, Information & Database

Data

Anything which is not necessarily meaningful to human being.

Raw facts/Un-processed information.



Defining Data, Information & Database

Examples of Data

The numbers 10, 22, and 119 are data

Some one can think these are lengths and other may think room numbers

Similarly, 1.5, 2.5, 31.5, ... **MAY** represent lengths of some iron rods

Defining Data, Information & Database

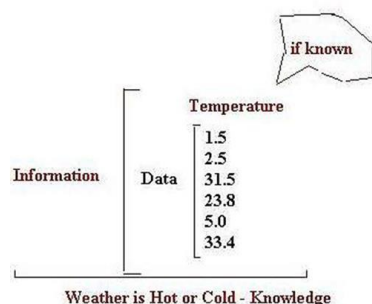
Information

Data Processed to reveal its meaning

Information is meaningful

In today's world, accurate, relevant and timely information is the key to good decision making

Good decision making is key to survival in today's competitive and global environment

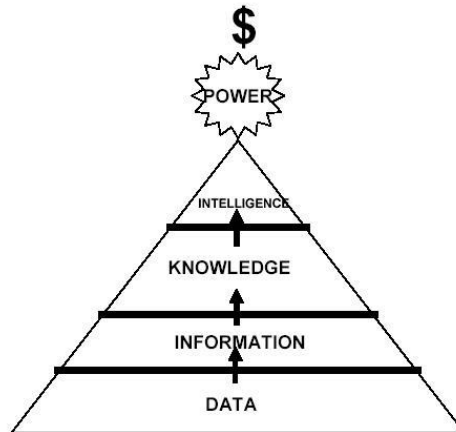


Defining Data, Information & Database

Examples of Information Software Engineer designed some system

May be monthly reports are not ok
Reports are giving wrong information, etc.

This means DESIGN is not ok



Defining Data, Information & Database

Aim of Data & Information



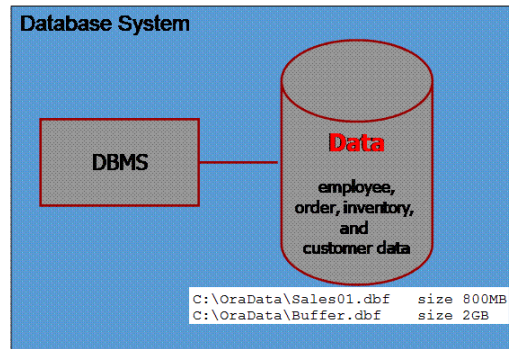
Defining Data, Information & Database

Who Builds Useful Information

Data Information

Defining Data, Information & Database

Database & DBMS



A database is collection of stored operational data used by application systems of some particular enterprise (C.J. Date)

TOPIC 02

Introduction to DBMS & its environment for DBA

Introduction to DBMS & its environment for DBA

DBMS, Role of DBA with Database

Examples of DBMS

Data Accessing using DBMS

Forms, Reports & Queries

DBMS Language

DBMS & its Sub Schemas

Role/ Jobs in DB Environment

Components of DBMS Environment

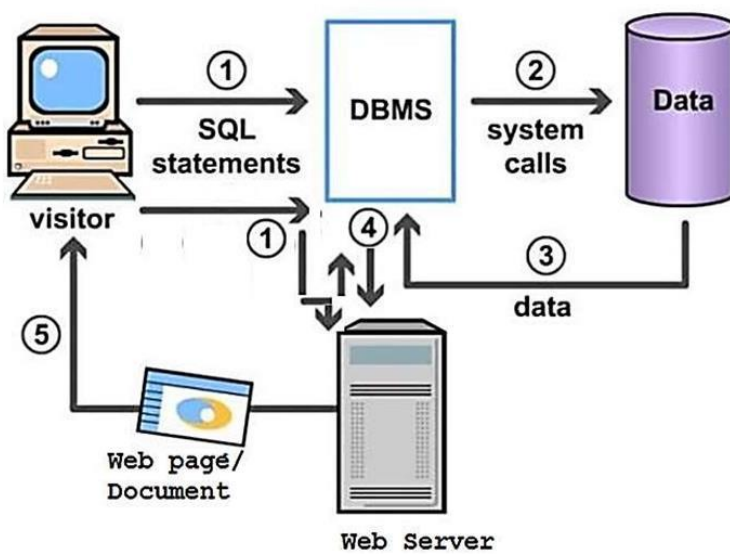
Interaction of DBMS with DB Files



Introduction to DBMS & its environment for DBA

Examples of DBMS

Oracle
 IBM DB2
 Ingress
 Teradata
 MS SQL Server
 MS Access
 MySQL etc.



Introduction to DBMS & its environment for DBA

Data Accessing using DBMS

tiers & 3-tiers Architectures

StudentNumber	StudentName	EmailAddress
100	Cooke	Cooke@OurU.edu
400	Greene	Greene@OurU.edu

PART

Number	PartName	SalesPrice	ROQ	QOH	Company Name	City	Country
1000	Cedar Shakes	\$22.00	100	200	Birdol Systems	Manchester	England
					ERS Systems	Vancouver	Canada
					Forest Supplies	Denver	US
2000	Garage Heater	\$1,750.00	3	4	Birdol Systems	Manchester	England
					ERS Systems	Vancouver	Canada
					Kyoto Importers	Kyoto	Japan
					Forest Supplies	Denver	US
3000	Utility Cabinet	\$55.00	7	3	Apex Manufacturing	Sydney	Australia

```
SQL> SELECT empno, ename, sal  
2 FROM EMP;
```

EMPNO	ENAME	SAL
7369	SMITH	800
7499	ALLEN	1600

Introduction to DBMS & its environment for DBA
Forms, Reports & Queries

Introduction to DBMS & its environment for DBA
DBMS Language

A Software Package that enables users to define, create, maintain, and control access to the database.

Data Definition Language (DDL) Data Manipulation Language (DML) Control access (DCL): Security, integrity, concurrent access, recovery, support for data communication, etc.

Utility services (DCL)

File import/export, monitoring facilities, code generator, report writer, etc.

Support Ad Hoc queries

Database Languages are DML, DDL, DCL in SQL.

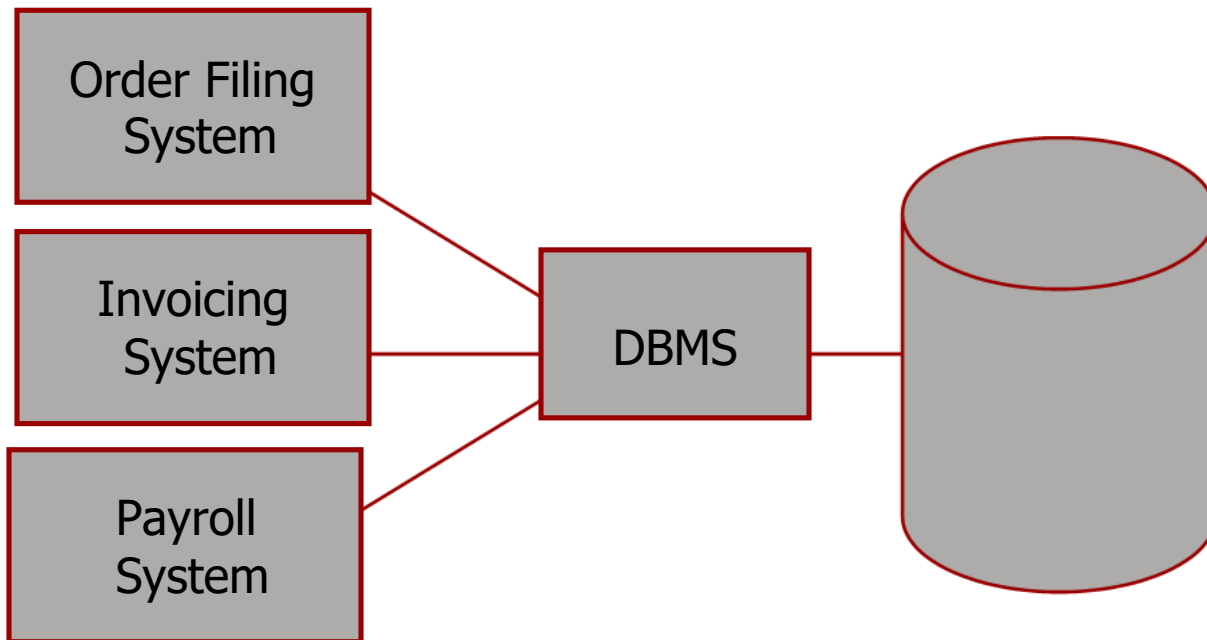
Introduction to DBMS & its environment for DBA

A software system that is used to create, maintain, and provide

Order Filing System

controlled access to user databases

DBMS and its Sub Schemas



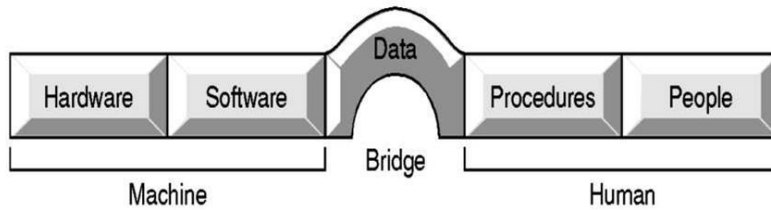
Contains employee, order, inventory, pricing, and customer data

DBMS manages data resources like an operating system manages hardware resources

Introduction to DBMS & its environment for DBA

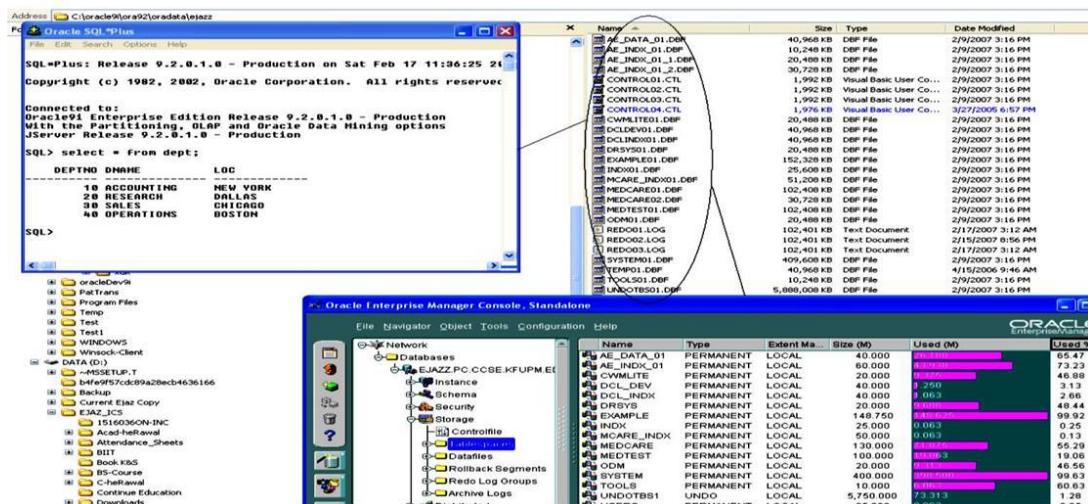
Roles/ Jobs in the Database Environment

- Data Administrator (DA)
- Database Administrator (DBA)
- Database Designers (Logical and Physical)
- Applications Programmer
- Database Developer
- Database Analyst
- End users (Naïve and sophisticated)



Introduction to DBMS & its environment for DBA

Components of DBMS Environment



Introduction to DBMS & its environment for DBA

Interaction of DBMS with DB Files

Introduction to Database Administration

TOPIC 03

Introduction to DBMS Components & Architecture

Introduction to DBMS Components & Architecture

DBMS Components & Architecture

Interaction of users with DBMS

ANSI Spark three level architecture

Multi user DBMS Architecture

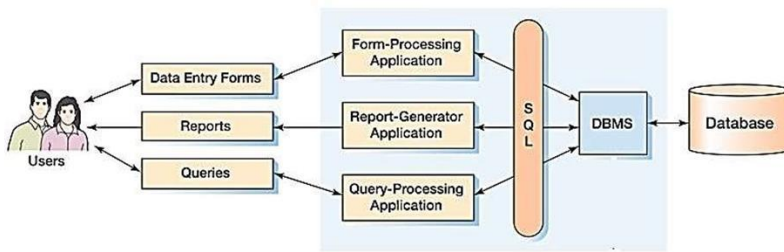
Teleprocessing

DBMS Client-Server Architecture (2- tiers)

DBMS Client-Server Architecture (3- tiers)

Alternate Client-Server Architectures

Paradigm of Database System



Introduction to DBMS Components & Architecture

Interaction of users with DBMS

Form Processing & Report Processing Applications are build by using VB, DOT Net or PHP programming. Reports can be developed by using Crystal Reports tool.

Query Processing can be managed by using vendors SQL tool or 3rd party tools such As TOAD, SQL Developer etc.

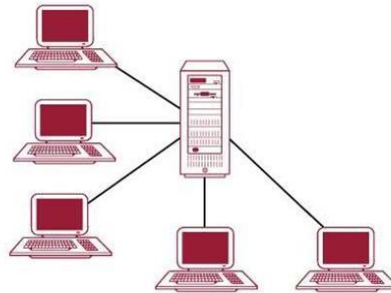
Introduction to DBMS Components & Architecture

Multi user DBMS Architecture

Teleprocessing

File-Server

Client-Server (2-tiers, 3 tiers)



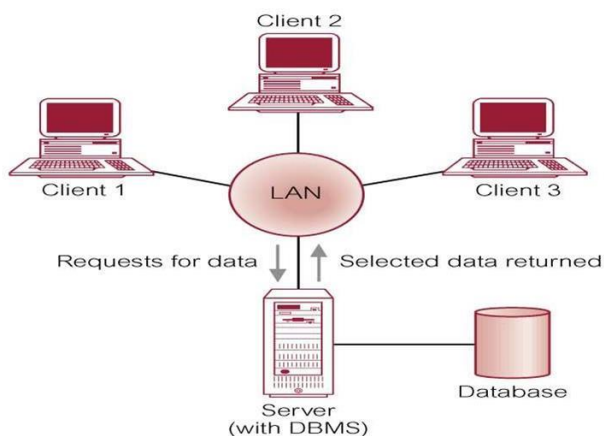
Introduction to DBMS Components & Architecture

Teleprocessing

Traditional architecture

Single mainframe with a number of terminals attached

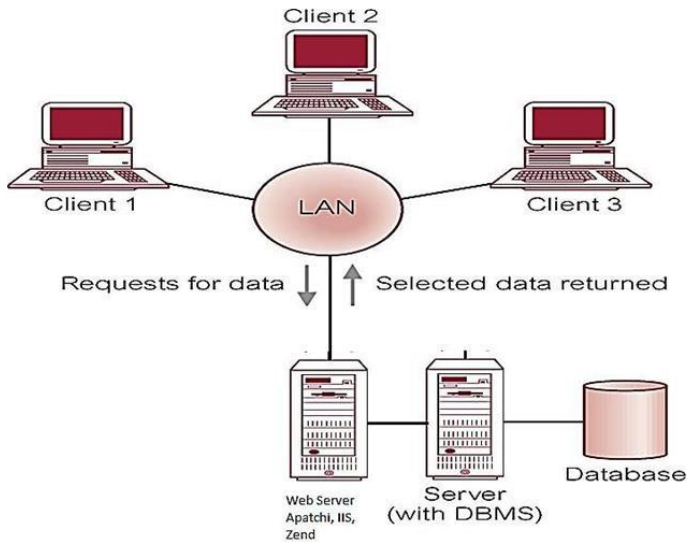
Trend is now towards downsizing



Introduction to DBMS Components & Architecture

DBMS Client-Server Architecture (2-tiers)

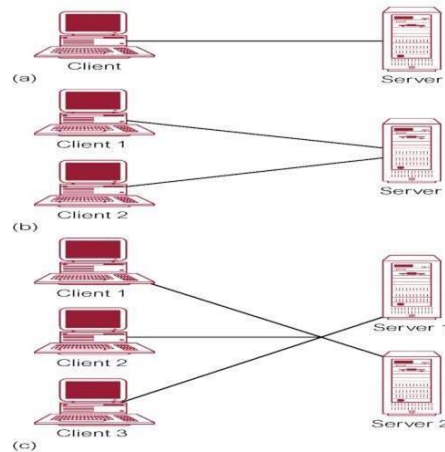
DB Connections can be managed by
ODBC, JDBC



Introduction to DBMS Components & Architecture

DBMS Client-Server Architecture (3-tiers)

DB Connections can be managed by using MS
ODBC, JDBC



Introduction to DBMS Components & Architecture

Alternate Client-Server Architectures

Introduction to DBMS Components & Architecture

Paradigm of Database System

Logical/ Conceptual Database Design

How should the transactional data be modeled (OLTP)?

Designing the data warehouse schema (OLAP)

Query Processing

Analysis queries are hard to answer efficiently

What techniques are available to the DBMS?

Introduction to DBMS Components & Architecture

Paradigm of Database System

Physical Database Design (DBA)

How should the data be organized on disk?

What data structures should be used?

Data Mining

What use is all this data?

Which questions should we ask our data warehouse OLAP?

Data Science/Big Data

Introduction to Database Administration

TOPIC 04

Introduction to Three Levels ANSI SPARC Architecture

Introduction to DBMS Three Levels Architecture

DBMS Three Levels Architecture

DBMS Three Levels Architecture

ANSI-SPARC Three-Level Architecture

Data Independence

Degrees of Data Abstraction

Schema & Instance

Schema Objects

Database State

Introduction to DBMS Three Levels Architecture

DBMS Three Levels Architecture

External Level

User's view of the database Describes that part of database that is relevant to a particular user

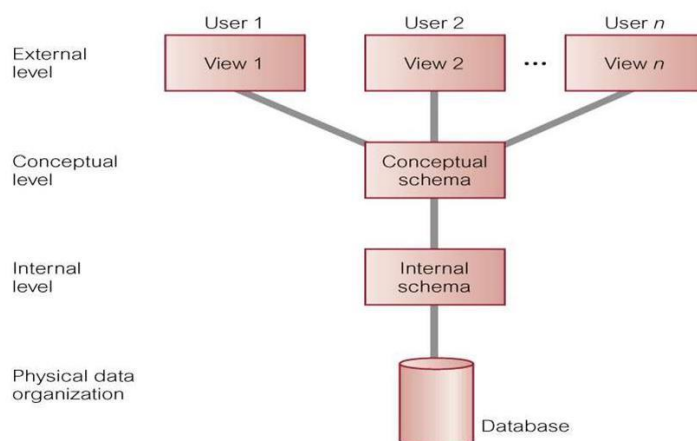
Conceptual Level

Community view of the database Describes what data is stored in database and relationships among the data

Internal Level

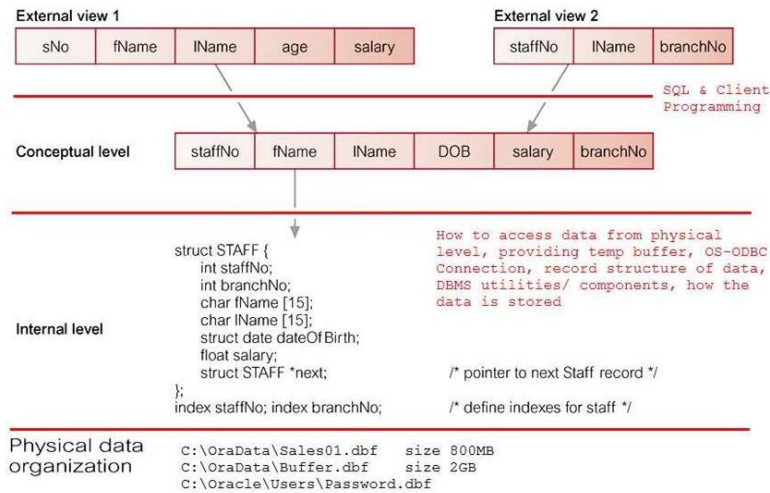
Physical representation of the database on the computer

Describes how the data is stored in the database



Introduction to DBMS Three Levels Architecture

ANSI-SPARC Three-Level Architecture



Introduction to DBMS Three Levels Architecture

ANSI-SPARC Three-Levels Architecture

Introduction to DBMS Three Levels Architecture

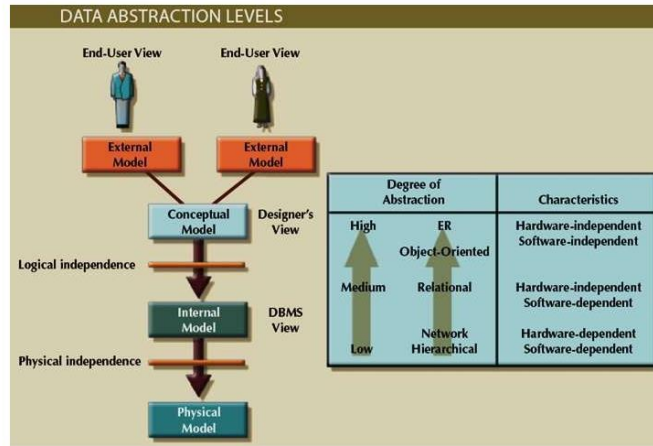
Data Independence

Logical Data Independence

Refers to immunity of external schemas to changes in conceptual schema

Conceptual schema changes (e.g. addition/removal of entities)

Should not required changes to external schema or rewrites of application programs



Introduction to DBMS Three Levels Architecture

Degrees of Data Abstraction

Introduction to DBMS Three Levels Architecture

```

Oracle SQL*Plus
File Edit Search Options Help

SQL*Plus: Release 9.2.0.1.0 - Production on Sat Feb 17 11:36:25 2006
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:
Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production

SQL> select * from dept;

  DEPTNO DNAME          LOC
-----
10 ACCOUNTING          NEW YORK
20 RESEARCH             DALLAS
30 SALES                CHICAGO
40 OPERATIONS           BOSTON

SQL>
  
```

Similar to types and variables in programming languages

Schema – the logical structure of the database

E.g., the database consists of information about a set of customers and accounts and the relationship between them

Analogues to type information of a variable in a program

Physical schema: database design at the physical level

Logical schema: database design at the logical level

Sub-Schema: there may be several of these at the view level, each a different view of database

Instance – the actual content of the database at a particular point in time

Analogous to the value of the variable

Schema & Instance

Introduction to DBMS Three Levels Architecture

Schema Objects

Tables

Indexes

Constraints

Sequences

Views

Clusters

Triggers

Links

Introduction to DBMS Three Levels Architecture

Schema Objects

User Defined **10**.Types **11**.Synonym **12**.Dimensions

13.Procedural Program codes in PL/SQL, Java, C etc.

STUDENT	Name	StudentNumber	Class	Major
	Smith	17	1	CS
	Brown	8	2	CS

COURSE	CourseName	CourseNumber	CreditHours	Department
	Intro to Computer Science	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Database	CS3380	3	CS

SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
	85	MATH2410	Fall	98	King
	92	CS1310	Fall	98	Anderson
	102	CS3320	Spring	99	Knuth
	112	MATH2410	Fall	99	Chang
	119	CS1310	Fall	99	Anderson
	135	CS3380	Fall	99	Stone

GRADE_REPORT	StudentNumber	SectionIdentifier	Grade
	17	112	B
	17	119	C
	8	85	A
	8	92	A
	8	102	B
	8	135	A

PREREQUISITE	CourseNumber	PrerequisiteNumber
	CS3380	CS3320
	CS3380	MATH2410
	CS3320	CS1310

Introduction to DBMS Three Levels Architecture

Database State

Introduction to Database Administration

TOPIC 05

Introduction to DBA Responsibilities & Tasks

Introduction to DBA Responsibilities & Tasks

DBA Roles and Responsibilities

Who is DBA?

DBA & Databases

Network Administrator

Application Developers

DBA's Tasks

DBA's Responsibilities

Introduction to DBA Responsibilities & Tasks

Specifying modifications of the database structure for an application
Introduction to DBA Responsibilities & Tasks

Application Developers

Relaying this information to a database administrator
Tuning the application during development
Establishing security measures for an application during development
Database Server Programming using Oracle PL/SQL
Introduction to DBA Responsibilities & Tasks

DBA's Tasks

Task 1: Evaluate the Database Server Hardware
Task 2: Install the Oracle Database Software
Task 3: Plan the Database **Task 4:** Create and Open the Database
Task 5: Back Up the Database **Task 6:** Enroll System Users
Task 7: Implement the Database Design
Introduction to DBA Responsibilities & Tasks

DBA's Tasks

Task 8: Back Up the Fully Functional Database
Task 9: Tune Database Performance **Task 10:** Download and Install Patches
Task 11: Roll Out to Additional Hosts
Introduction to DBA Responsibilities & Tasks

DBA's Responsibilities

Installing and upgrading the Oracle Database server and application tools
Allocating system storage and planning future storage requirements for the database system
Creating primary database storage structures (tablespaces) after application developers have designed an application
Introduction to DBA Responsibilities & Tasks

DBA's Responsibilities

Creating primary objects (tables, views, indexes) once application developers have designed an application

Modifying the database structure, as necessary, from information given by application developers

Introduction to DBA Responsibilities & Tasks

DBA's Responsibilities

Enrolling users and maintaining system security

Ensuring compliance with Oracle license agreements

Controlling and monitoring user access to the database

Monitoring and optimizing the performance of the database

Introduction to DBA Responsibilities & Tasks

DBA's Responsibilities

Planning for backup and recovery of database information

Maintaining archived data on tape

Backing up and restoring the database

Contacting Oracle for technical support

END

Introduction to Database Administration

TOPIC 06

Overview of Physical Database Design

Overview of Physical Database Design

Physical Database Design

Physical level in ANSI SPARC Arch.

Purposes of Physical Database Design

Factors that influence Physical Database Design

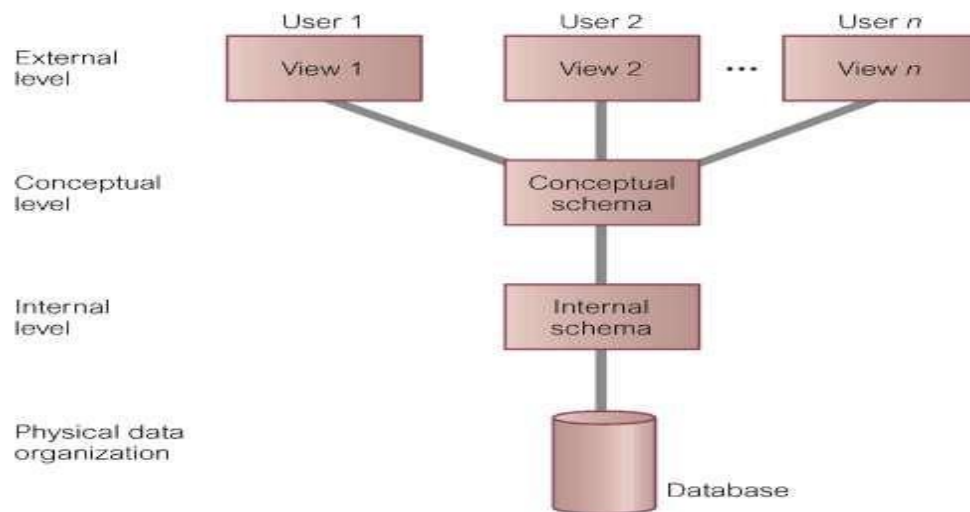
Design decisions about Indexing

Performance & Tuning

Tuning Indexes

Overview of Physical Database Design

Physical Level in ANSI SPARC Architecture



Ref. DBA-4 (4)

Overview of Physical Database Design

Purposes

Meeting the expectations of Database Designer for the database, following are two main purposes of Physical Database Design for a DBA.

Managing Storage Structure for database or DBMS

Performance & Tuning

Overview of Physical Database Design

Factor (A): Analyzing the database queries and transactions

For each query, the following information is needed.

The files that will be accessed by the query

The attributes on which any selection conditions for the query are specified;

The attributes on which any join conditions or conditions to link multiple tables or objects for the query are specified

The attributes whose values will be retrieved by the query.

Overview of Physical Database Design

Factor (A): Analyzing the database queries and transactions

For each update transaction or operation, the following information is needed.

The files that will be updated;

The type of operation on each file

(insert, update or delete);

The attributes on which selection conditions for a delete or update operation are specified;

The attributes whose values will be changed by an update operation.

Overview of Physical Database Design

Factor (B): Frequency with Queries and Transactions

The expected frequency information, along with the attribute information collected on each query and transaction, is used to compile a cumulative list of expected frequency of use for all the queries and transactions.

It is expressed as the expected frequency of using each attribute in each file as a selection attribute or join attribute, over all the queries and transactions.

Overview of Physical Database Design

Factor (C): Time constraints of queries & transactions

Performance constraints place further priorities on the attributes that are candidates for access paths.

The selection attributes used by queries and transactions with time constraints become higher-priority candidates for primary access structure.

Overview of Physical Database Design

Factor (D): Expected frequencies of update operations

A minimum number of **access paths** should be specified for a file that is updated frequently.

Overview of Physical Database Design

Factor (E): Uniqueness constraints on attributes

Access paths should be specified on all candidate key attributes

or set of attributes

that are either the primary key or constrained to be unique.

Overview of Physical Database Design

Design Decisions About Indexing

Whether to index an attribute?

What attribute or attributes to index on?

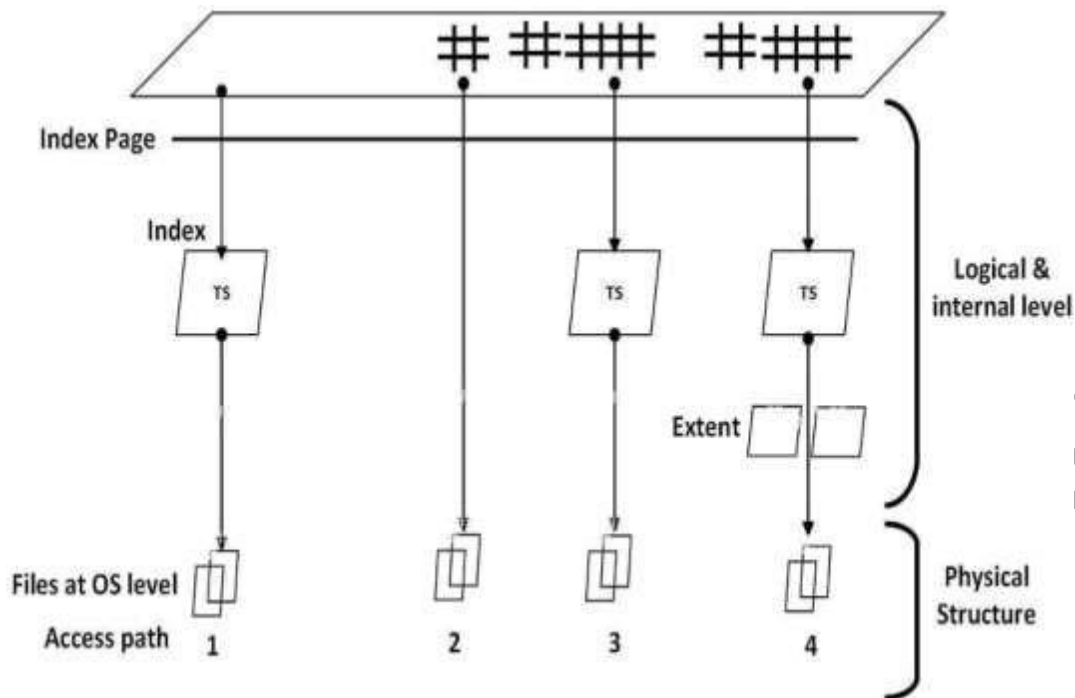
Whether to set up a clustered index?

Whether to use a hash index over a tree index?

Whether to use dynamic hashing for the file?

Overview of Physical Database Design

Access Path & Storage



END

Introduction to Database Administration

TOPIC 07

Introduction to Database Tuning

Introduction to Database Tuning

Database Tuning

Tuning & goals

Tuning indexes

Overview of database tuning

Problems in tuning

Dynamically changes processing requirements

Changes in Database Design

Tuning queries

Introduction to Database Tuning

Tuning & Goals Tuning:

The process of continuing to revise/adjust the physical database design by monitoring resource utilization as well as internal DBMS processing to reveal bottlenecks such as contention for the same data or devices.

Introduction to Database Tuning

Tuning & Goals Goal:

To make application run faster

To lower the response time of queries/ transactions

To improve the overall throughput of transactions

Introduction to Database Tuning

Tuning Indexes

Reasons to tuning indexes

Certain queries may take too long to run for lack of an index;

Certain indexes may not get utilized at all;

Certain indexes may be causing excessive overhead because the index is on an attribute that undergoes frequent changes

Introduction to Database Tuning

Tuning Indexes

Options to tuning indexes

Drop or/and build new indexes

Change a non-clustered index to a clustered index (and vice versa)

Rebuilding the index

Introduction to Database Tuning

Overview of Database Tuning

Statistics obtained from monitoring:

Storage statistics

I/O and device performance statistics

Query/transaction processing statistics

Locking/logging related statistics
Index statistics
Introduction to Database Tuning

Overview of Database Tuning

Statistics internally collected in DBMSs:

Size of individual tables

Number of distinct values in a column

The number of times a particular query or transaction is submitted/executed in an interval of time

The times required for different phases of query and transaction processing

Introduction to Database Tuning

Problems in Tuning

How to avoid excessive lock contention?

How to minimize overhead of logging and unnecessary dumping of data?

How to optimize buffer size and scheduling of processes?

How to allocate resources such as disks, RAM and processes for most efficient utilization?

Introduction to Database Tuning

Dynamically Changed Processing Requirements

Need to be addressed by making changes to the conceptual schema if necessary and to reflect those changes into the logical schema and physical design.

Introduction to Database Tuning

Changes in Database Design

Existing tables may be joined (de-normalized) because certain attributes from two or more tables are frequently needed together.

For the given set of tables, there may be alternative design choices, all of which achieve 3NF or BCNF. One may be replaced by the other.

Introduction to Database Tuning

Changes in Database Design

A relation of the form $R(\underline{K}, A, B, C, D, \dots)$ that is in BCNF can be stored into multiple tables that are also in BCNF by replicating the key K in each table.

Attribute(s) from one table may be repeated in another even though this creates redundancy and potential anomalies.

Apply horizontal partitioning as well as vertical partitioning if necessary.

Introduction to Database Tuning

Tuning Queries

Indications for tuning queries

A query issues too many disk accesses

The query plan shows that relevant indexes are not being used.

Introduction to Database Tuning

Tuning Queries

Typical instances for query tuning

Many query optimizers do not use indexes in the presence of arithmetic expressions, numerical comparisons of attributes of different sizes and precision, NULL comparisons, and sub-string comparisons.

Indexes are often not used for nested queries using IN;

Introduction to Database Tuning

Tuning Queries

Some *DISTINCT*s may be redundant and can be avoided without changing the result.

Unnecessary use of temporary result tables can be avoided by collapsing multiple queries into a single query unless the temporary relation is needed for some intermediate processing.

If multiple options for join condition are possible, choose one that uses a clustering index and avoid those that contain string comparisons.

Introduction to Database Tuning

Tuning Queries

In some situations involving using of correlated queries, temporaries are useful.

The order of tables in the *FROM* clause may affect the join processing.

Some query optimizers perform worse on nested queries compared to their equivalent un-nested counterparts.

Many applications are based on views that define the data of interest to those applications.

Sometimes these views become an overkill.

Introduction to Database Administration

TOPIC 08

Concepts of Keys and Super keys

Concepts of Keys & Super Keys

Keys & Super Keys

Concepts of Keys

Super Keys

Example – 1

Example – 2

Concepts of Keys & Super Keys

Concepts of Keys

A **key** is a combination of one or more columns that is used to identify rows in a relation



A **composite key** is a key that consists of two or more columns

Concepts of Keys & Super Keys

Super Keys

A super key is a combination of columns that uniquely identifies any row within a relational database management system (RDBMS) table.

Roll#	FirstName	LastName	Address	City	NIC#	Deptno

12i-2565	---					CS
	Ali					CS
	Ali					EE

{Roll#}, {Roll#, NIC#}, {NIC#}, {Roll#, FirstName, LastName}

{Roll#, FirstName, LastName, Address, City, NIC#, Deptno}

Concepts of Keys & Super Keys

Super Keys Example - 1

Roll Number	First Name	Last Name
CSU0001	Shabbir	Bhimani
CSU0002	Jain	Rao
CSU0003	Nitin	Uday

Now we have the following as super keys

Roll Number	First Name	
Roll Number	First Name	Last Name

Concepts of Keys & Super Keys

Super Keys Example - 2

Following table consists of four columns EmployeeID, Name, Job, DeptID

Examples of superkeys in this table would be

{employeeID, Name}, {employeeID, Name, job}, and

{employeeID, Name, job, departmentID}

In a real database we don't need values

for all of those columns to identify a row

Concepts of Keys & Super Keys

Super Keys Example - 2

We only need, per our example, the set {EmployeeID}.

This is a minimal superkey

So, employeeID is a candidate key.

EmployeeID can also

uniquely identify the tuples.

TOPIC 09

Candidate Keys with Examples

Candidate Keys with Examples

Candidate Keys

Candidate Key Definition

Example (Invoice)

Example (Bank, PTCL, NADRA)

Candidate Keys with Examples

Candidate Key Definition

A **candidate key** is a key that determines all of the other columns in a relation

Candidate key columns help in searching **fewer duplicated** or unique records.

Examples

In PRODUCT relation Prod# is a candidate key

Prod_Name is also a candidate key In ORDER_PROD

(OrderNumber, Prod#) is a candidate key

(Candidate key can be called as Alternate Key)

Candidate Keys with Examples

Candidate Key Example Invoice (Buying Items)

Items are sold to Customers

Customer should buy Items with Invoice

Same items can be sold on many invoices

Invoice# and Item# will identifying exact record, what other columns are required

If some one tells you inv# and Qty, can you find exact product

Candidate Keys with Examples

Candidate Key Examples

For a Bank as an Enterprise: **AccountHolder** (or **Customer**)

ACC#, Fname, Lname, DOB, **CNIC#**, Addr,
City, **TelNo**, **Mobile#**, **DriveLic#**

For PTCL: **Customer** (single telno holders) Consumer#, Fname, Lname, DOB, **CNIC#**,
Addr, City, **TelNo**, **Mobile#**

END

For NADRA: Citizen (**CNIC#**, Fname, Lname, FatherName, DOB, **OldCNIC#**, PAddr, PCity,
TAddr, TCity, **TelNo**, **Mobile#**)

Introduction to Database Administration

TOPIC 10

Candidate Keys with Examples

Candidate Keys with Examples

Candidate Keys Continued

Candidate Key examples

Exercises

Candidate key for search purpose

Candidate Keys with Examples

Candidate Key Examples

Example-1: Branch (branch-name, assets, branch-city)

Candidate Key: {branch-name} Example-2: Customer (cname, natid,
address, city, telno)

cname, address, city can be duplicated individually and cannot determine a record.

Candidate Keys with Examples

Candidate Key Examples

The following combinations distinguish customer records or tuples.

{cname, telno}, {natid}

{natid, cname}

As {natid} $\square$ {natid, cname}, then

{natid, cname} is not candidate key and {natid} is a candidate key

Example-3: Employee(empno, name, birth_date, address, city, telno, citizenship_id)

empno, telno, citizenship_id are possible candidate keys

Candidate Keys with Examples

Exercises

Course (cid, cname, deptno) Semester (sid, syear, startdate) ClassAllocation (sid, cid, sec#, building#, room#)

Identify candidate keys in each of the above relations

Candidate Keys with Examples

Candidate Key for Search purpose

The screenshot shows a software interface for managing medicines. At the top, there's a search form with fields for STOCK_NUMBER, BRAND_NAME, GENERIC_NAME, MEDICINE_TYPE, STOCK_TYPE, MG_ML, BARCODE, and DOSAGE_DESC. There are also checkboxes for AVAILABLE_STATUS and UNIT_TYPE. A 'Find' button is present. Below the search form, there's a table with columns: stock_no, brand_name, generic_name, medicine_type, stock_t, MG_ML, Barcode, Dosage_desc, available, m_name, and manu. The table contains several rows of medicine data. A watermark '97869@gmail.com' is visible diagonally across the image.

stock_no	brand_name	generic_name	medicine_type	stock_t	MG_ML	Barcode	Dosage_desc	available	m_name	manu
3902982	CALPOL 6 PLUS 90ML 1S	PARACETAMOL		01	MG			Y		1
3902985	DISPOL 90ML	PARACETAMOL		01	MG			Y		73
3901844	SAMEROL-N FORTE	PARACETAMOL + ORPHANADRIN	PK/10TAB		MG			Y		87
3901850	SAMEROL-N	PARACETAMOL + ORPHANADRIN	PK/72TAB		MG			Y		87
3903400	PANADOL TAB 500MG	PARACETAMOLE		01	MG			Y		1
3903401	PANADOL CF 500MG 100S	PARACETAMOLE		01	MG			Y		1
3903283	FEBROL DS 60ML	PARACETAMOL	BOT/120MG	01	MG					35
3901595	NUBEROL	PARACETAMOL BP & ORPHENAD	PK/100TAB		MG			Y		32
3901596	NUBEROL FORTE	PARACETAMOL BP & ORPHENAD	PK/15TAB		MG			Y		32
3900455	CALPOL SYP 100ML 1S	PARACETAMOL		01	ML			Y		1
3903285	TEMPRAMINE 60ML	PARACETAMOL		01						116
3902971	TEMPOL 60ML	PARACETAMOL	BOT/120MG		MG			Y		117
3902972	TEMPOL 6+ 60ML	PARACETAMOL	BOT/		MG			Y		117
3902973	FEBROL 60ML	PARACETAMOL	BOT/120MG	01	MG			Y		35
3902974	PANADOL SYRUP 100ML 1S	PARACETAMOL		01	MG			Y		1

DBA can create Index on searching attributes (candidate Keys)

Introduction to Database Administration

TOPIC 11

Primary Key

Primary Key

Primary Keys

Primary key definition

An alternate definition

Primary key examples & issues

Primary Key

Primary Key Definition

A **primary key** is a candidate key selected as the primary means of identifying rows in a relation:

There is one and only one primary key per relation

The primary key is NOT NULL, UNIQUE

The ideal primary key is short, numeric(alpha), indexed, fixed length and never changes

Key that has its ownership of an

Enterprise

The primary key may be a **composite key**

Primary Key

Primary key

An Alternate Definition

A primary key is a minimal identifier that is used to identify tuples uniquely.

This means that no subset of the primary key is sufficient to provide unique identification of tuples.

NULL values is not allowed in primary key attribute.

Example:

Primary key - Example & Issues

STU_ID

(Practically PK must have UNIQUE, NOT NULL, INDEXO on an adribute)

S^{TUDENT}

STU_ID	NAME	CID_ID	DOB	DOB DEPTCODE
--------	------	--------	-----	-----------------

Although, candidate keys are STU_ID, CIT_ID. Organization ITSELF issues its Dwn identifier PK. PK is a property of an organization. Driving license# is a property of Ministry of Motors/ transportation, it must not be used PK in university or other organizations. It is a candidate key for other organization for search puposes,

Note: It is not recommended to create PKs for columns like Quantity, Salary, Price and ^{column} that has datatype D^{ATE}

Introduction to Database Administration

TOPIC 12

Primary Keys Examples

Primary Keys Examples

Primary Keys

Primary Key – student

Primary Key - More Examples

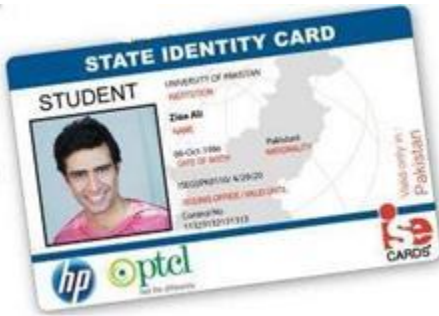
Primary Keys Examples

Primary key - Student

Example-1:

STUDENT(StuID, FirstName, FamilyName, DOB, ...)

Example-2: Building (B#, BName, Location, Region),



B# is a primary key. Although, BName is unique, not null but it is not short.

Primary Keys Examples

Primary key - More Examples

Example-3: Customer (cname, citizenid, address, city, telno)

This relation indicates the information about personal details. There is a chance that cname is duplicated, some may have citizenid and telno as null. This forces us to introduce new a attribute such as cust# that would be a primary key.

Customer (cust#, cname, citizenid, address, city, telno)

Primary Keys Examples

Primary key - More Examples

Example-4:

BankBranch(Branch-Name, City, TotalAsset)

What will be a PK that you will suggest?

What is a candidate key?

Which attribute is unique?

There may be many branches in one city then finalize this relation with possible constraints

Introduction to Database Administration

TOPIC 13

Primary Keys Examples

Primary Keys Examples

Primary Keys

Primary key instances

Fixed length examples

Indexing

Primary Keys Examples

Primary key Instances

512235665 Social Security Number 610112654 (By Birth Asian, European, HomeLand)

200338900 Student ID (With Registration year)

LHR051 Product ID (with Manufacturing Loc

31266885 Bank Acc# (followed by



Branch Code)



Primary Keys Examples

Primary key Instances

Serial# starts with 1 and incremented by 1 005384 Company has many branch codes 005

007004 invoices issued from branches

Primary Keys Examples

Primary key Fixed Length

After clicking on  compare your screen with the one below.

St	Item A	I	Material	Short Text	Quantity	Unit	C	Delivery Date	Mati Group	Plant	Stor. Loc.	PGr
	10		CHLK1000	Chain Lock	200	EA	0	09/13/2017	Utilities	DC Miami	Trading G.	

Then, click on  to save your purchase requisition. The system will create a unique document number.  Purchase requisition number 0010000000 created

Click on the exit icon  to return to the SAP Easy Access screen

Vendor Account Number (1) 141 Entries found

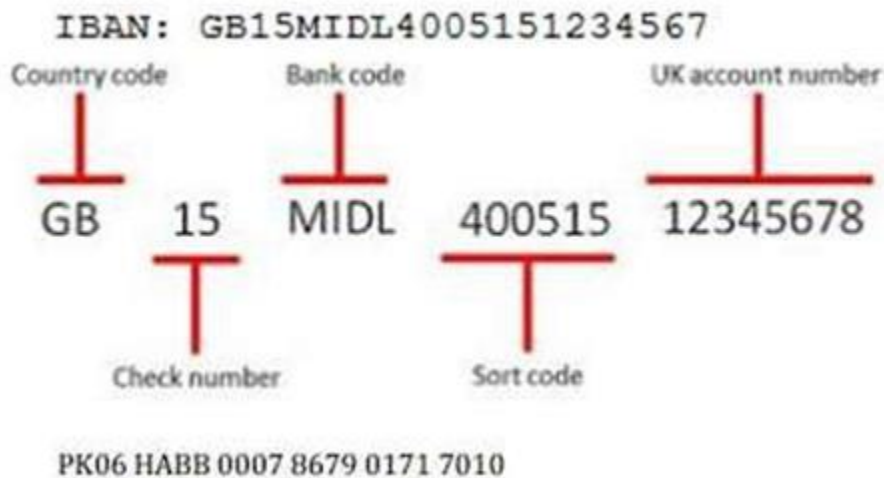
A: Vendors (General) I: Vendors by Country/Company Code K: Ve...

SearchTerm	Cty	PostalCode	City	Name 1	Vendor
055	US	68516	LINCOLN	MID-WEST SUPPLY	125005
055	US	68528	LINCOLN	MID-WEST SUPPLY	125094
056	US	68528	LINCOLN	MID-WEST SUPPLY	125096
056	US	68528	LINCOLN	MID-WEST SUPPLY	125100
056	US	68528	LINCOLN	MID-WEST SUPPLY	125112
057	US	68528	LINCOLN	MID-WEST SUPPLY	125115
059	US	68528	LINCOLN	MID-WEST SUPPLY	125128

Encircled Fixed Length Key Identifiers

Primary Keys Examples

Primary key - Fixed Length



Primary Keys Examples

```

Table: PATIENT_VISIT
Index: Yes, PK
Test Query execution time: less than 1 second

Table: PATIENT_VISIT1
Index: NO
Test Query execution time: 12 seconds

SQL> select count(*) Total_Records
      2 from PATIENT_VISIT;

      Total_Records
-----|
      625721

Elapsed: 00:00:00.04

```

```

SQL> select PATIENT_ID, max(PATIENT_VISITNO)
      2 from PATIENT_VISIT1
      3 where PATIENT_VISITNO = (select max(PATIENT_VISITNO)
      4 from PATIENT_VISIT1
      5 where DEPEND_SNO='0')
      6 group by PATIENT_ID
      7 having count(*)>=1;

```

```

PATIENT_ID MAX(PATIENT_VISITNO)
-----|-----
9999993      1316

```

Elapsed: 00:00:11.54

PRIMARY KEY (PATIENT_ID, DEPEND_SNO, PATIENT_VISITNO)

Primary key - Indexing

Introduction to Database Administration

TOPIC 14

Primary Key Formats

Primary Keys Formats

Primary Keys

Primary key Indexing

Basics of indexing

Indexing on DBMS

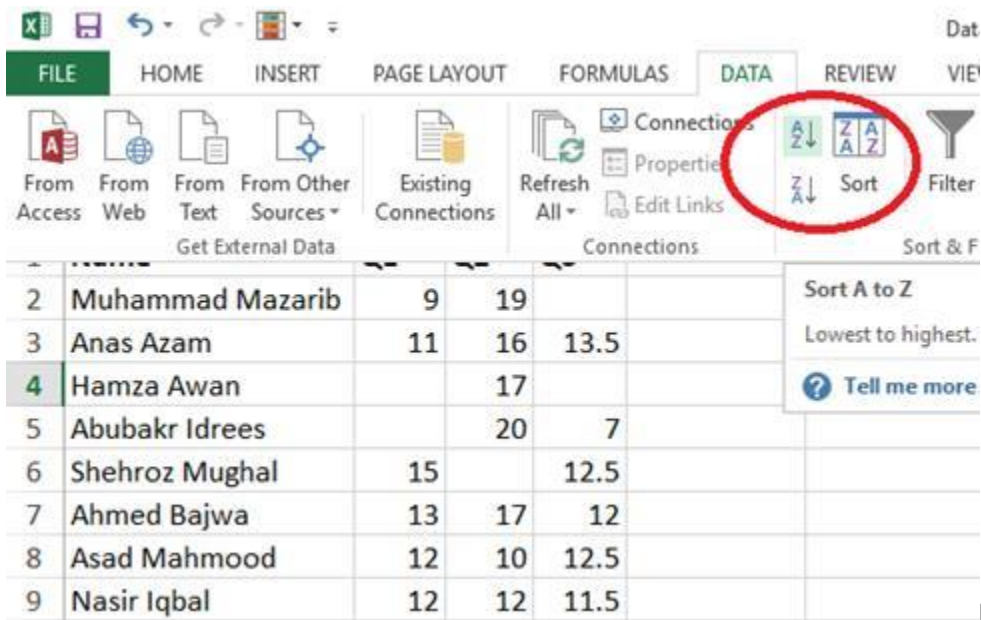
Who will decide?

Primary Keys Formats

Basics of Indexing

Paging a key's data to access fast.

It may includes sorting from A to Z or Z to A.



Indexing is applied on non-key columns where mostly search is required
 Primary Keys Formats

Primary key – Indexing on DBMS

```
Table: PATIENT_VISIT
Index: Yes, PK
Test Query execution time: less than 1 second

Table: PATIENT_VISIT1
Index: NO
Test Query execution time: 12 seconds

SQL> select count(*) Total_Records
      2  from PATIENT_VISIT;

      Total_Records
      -----|
      625721

Elapsed: 00:00:00.04
```

```
SQL> select PATIENT_ID, max(PATIENT_VISITNO)
      2  from PATIENT_VISIT1
      3  where PATIENT_VISITNO = (select max(PATIENT_VISITNO)
      4    from PATIENT_VISIT1
      5    where DEPEND_SNO='0')
      6  group by PATIENT_ID
      7  having count(*)>=1;
```

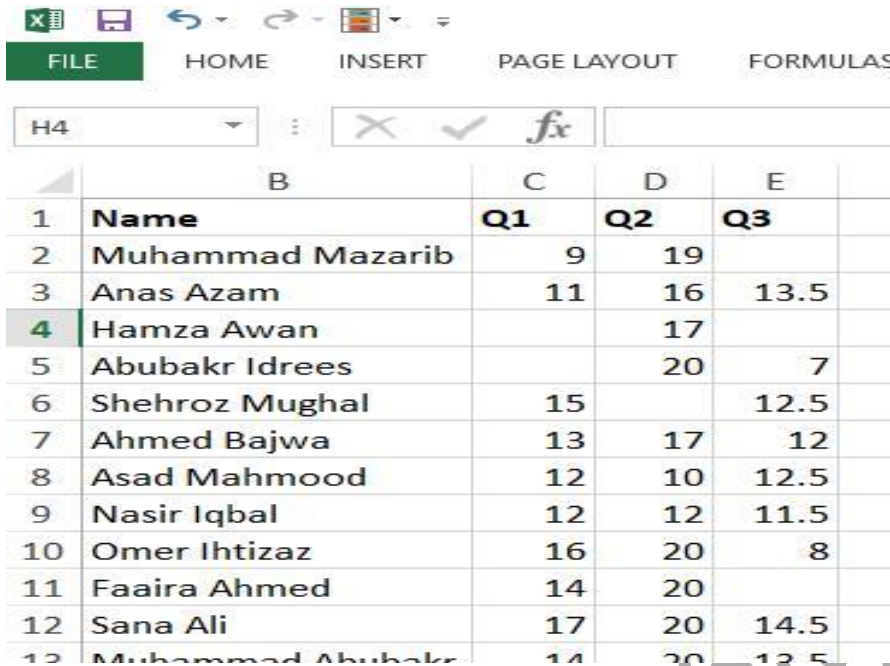
```
PATIENT_ID MAX(PATIENT_VISITNO)
-----|-----
99999993          1316

Elapsed: 00:00:11.54
```

PRIMARY KEY (PATIENT_ID, DEPEND_SNO, PATIENT_VISITNO)

Primary Keys Formats

Primary key – Who will decide?



The screenshot shows an Excel spreadsheet with a table of student names and their scores on three quizzes (Q1, Q2, Q3). The table is located in the range B1:E13. The 'Name' column is in B, 'Q1' is in C, 'Q2' is in D, and 'Q3' is in E. The rows are numbered 1 to 13. The student 'Hamza Awan' is highlighted in row 4.

	B	C	D	E
	Name	Q1	Q2	Q3
1	Muhammad Mazarib	9	19	
2	Anas Azam	11	16	13.5
3	Hamza Awan		17	
4	Abubakr Idrees		20	7
5	Shehroz Mughal	15		12.5
6	Ahmed Bajwa	13	17	12
7	Asad Mahmood	12	10	12.5
8	Nasir Iqbal	12	12	11.5
9	Omer Ihtizaz	16	20	8
10	Faaira Ahmed	14	20	
11	Sana Ali	17	20	14.5
12	Muhammad Abubakr	14	20	12.5

Roll# is issued by some University or college Format or instance, meaningful column for PK
Introduction to Database Administration

TOPIC 15

Surrogated Keys

Surrogated Keys

Definition

Examples

Exercises

Surrogated Keys

Surrogated Key Definition

A **surrogate key** as an artificial column added to a relation to serve as a primary key:

DBMS supplied

Short, numeric and never changes – an ideal primary key!

Has **artificial values** that are meaningless to users

Normally hidden in forms and reports

Surrogated Keys

Example

RENTAL_PROPERTY **without** surrogate key:

RENTAL_PROPERTY (Street, City, State/Province, Zip/PostalCode, Country, Rental_Rate)

RENTAL_PROPERTY with surrogate key:

RENTAL_PROPERTY (PropertyID,



Street, City, State/Province, Zip/PostalCode, Country, Rental_Rate)

Surrogated Keys

Exercises

Lets solve following examples

When to put up a complaint by using phone?

ComplaintRef# is normally generated

Needs attributes?

End

Introduction to Database Administration

TOPIC 16

Surrogated Keys Examples

Surrogated Keys Examples

Surrogated Keys Examples

More Examples

Insurance & ATM Transaction

Keys columns with order

Interesting example

Surrogated Keys Examples

More Examples

ATMTransaction

(Card#, Serial#, Amount, DrawDate)

Need more attributes

Machine#, Type, Status, LimitBalance

Surrogated Keys Examples

More Examples

InsurancePaid

(Policy#, PaidYear, PaidDate, Amount)

Need more attributes

DueDate, DueAmount, Balance, Charges

Surrogated Keys Examples

Keys with order in columns

<u>PatientID</u>	<u>Visit#</u>	Visit Date
558741	1	...
558741	2	
558741	3	
...		
589210	1	
589210	2	
599211	1	
599211	2	
...		

Better Solution
using composite keys

<u>PatientID</u>	<u>Visit#</u>	Visit Date
558741	1	...
558741	2	
558741	3	
589210	4	
589210	5	
599211	6	
599211	7	
...	...	

830000 and more

TOPIC 17

Surrogated Keys Examples

Interesting example

Lab Requisitions - drjaved

REQId	REQ Desc	Status
1	HEMATOLOGY	

Add Update Remove

Test Description	Ref. Range
1: HEMATOLOGY:Y	
2: BIOCHEMISTRY:Y	
3: SEROLOGY:Y	
4: HORMONES:Y	
5: ANTENATAL:Y	
6: URINE:Y	
7: STOOL:Y	
8: MICROBIOLOGY:Y	
9: SEMEN:Y	
10: MEDICAL CHECK-UPS:Y	
11: OTHERS:Y	

SubReqId 22 ReqId

Add Update Remove

SubLab req. id	Description	ReqId	Description:	Ref. Range
1: CBC		20	Hb : M 15.0 ± 2.0; F 13.5 ± 2.0 gm/dl	
2: Coagulation Studies		21	WBC : 4.0 – 10.0 x 10 ⁹ /L	
3: Others Hematology		22	Differential :	
		23	Platelets : 140 – 400 x 10 ⁹ /L	
		24	RBC : M 4.6 – 6.5; F 3.9 – 5.6 10 ¹² /L	
		25	Hct : M 0.39 – 0.50; F 0.34 – 0.44	
		26	MCV : M 81 -94; F 79 -91 fl	
		27	MCH : M 27 -32; F 25 -31 pg	
		28	MCHC : M 31 -35; F 30 – 34%	
		29	Reticulocytes : < 2%	

REQId SubReqId Desc

1 CBC

Add Update Remove

03/11/2013 11:46:29 AM

Total Records: 11

LABREQ	LABREQ_SNODESC	LABREQ
38	Band Cell	22
39	Neutrophil	22
40	Lymphocyte	22
41	Monocyte	22
42	Eosinophil	22

Introduction to Database Administration

Surrogated Keys Examples

Surrogated Keys Examples

Surrogated Keys

Solving example-1

Solving example-2

Surrogated Keys Examples

Solving Example-1

Let us choose

Invoice, when buying some items

Surrogated Keys Examples

Solving Example-2

Let us discuss

We visit some wedding hall to book some event.

How staff of hall will store event's details? Which columns are required.

Introduction to Database Administration

TOPIC 18

Surrogated Keys Examples

Surrogated Keys Examples

Surrogated Keys

Solving example-1

Solving example-2

Comparisons of Keys

Surrogated Keys Examples



Solving Example-1 (Saving blogs)

Surrogated Keys Examples

Solving Example-2

Let's us assume a loan of Rs. 50,000

How to decide which columns are required to fill up to keep track record of installments?

Surrogated Keys Examples

Comparisons of keys

Let us discuss

What is difference between PK & surrogated key

What is difference between PK & unique key

Give three more examples of surrogated keys from real-life.

End

Introduction to Database Administration

TOPIC 19

Foreign Keys

Foreign Keys

Foreign Keys

Definition

Example-1

Example-2

Foreign Keys

Definition

A **foreign key** is an attribute that refers to a primary key of same or different relation to form a link (constraint) between the relations:

A foreign key can be a single column or a composite key

The term refers to the fact that key values are *foreign* to the relation in which they appear as foreign key values

Ideally Data type, Length/ Size of FK and referring PK must be same

Foreign Keys

Example-1

DEPARTMENT (DeptID, DepartmentName,
BudgetCode, ManagerName)

EMPLOYEE (EmpNumber, EmpName, DeptID)

Name of FK column may be different from the
name of referencing PK



End

Foreign Keys

Example-2

CUSTOMER(CustId, Name, Address, Telno, Email)

ORDERS(Invoice#, InvType, InvDate,

CustomerId)

The screenshot shows a software window titled 'Invoice'. It contains the following information:

Lloyds Chemist
Street 87 G-13/1 Islamabad
(051)230 6867

Inv#: 3 Date: 11/29/2013

Code	Med Description	Qty	Price	Amount
3906020	PANADOL	1	12.5	12.5
3900001	ALRIAL DROPS	2	20	40
3906020	PANADOL	2	12.5	25
3900001	ALRIAL DROPS	3	20	60
Total:				137.5
Disc:				6
Net Total:				129.25

Refrigerator items are not refundable & exchangeable
Please check all items as per your prescription Sig. :

Pages: 1

Does Invoice# always contain serial numbers?



Introduction to Database Administration

TOPIC 20

Foreign Keys Examples
Foreign Keys Examples

Foreign Key Examples

Relationship details
FK & Recursive Relationship
Logical link between tables

Foreign Keys Examples

Relationship details



Deptno is a PK in DEPT table*

Empno is a PK in EMP table

Deptno is a FK in EMP table*

(it references to an attribute Deptno of DEPT table)

NOTE: An attribute within one relation that matches the key (Primary/same) in the same relation, is called recursive relation

Foreign Keys Examples

DEPT : Table

Deptno	Dname	Loc
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON
50	R & D	TORONTO

Record: 6 of 6

FK or Recursive Relationship

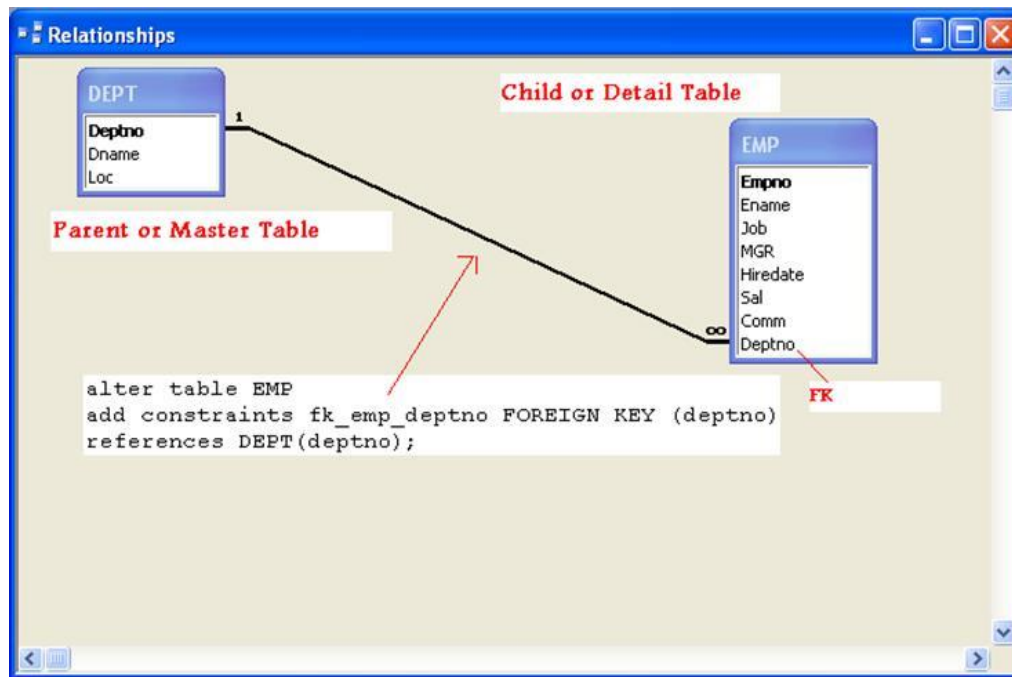
EMP : Table

Empno	Ename	Job	MGR	Hiredate	Sal	Comm	Deptno
7369	SMITH	CLERK	7902	17-Dec-80	800		20
7499	ALLEN	SALESMAN	7698	20-Feb-81	1600	300	30
7521	WARD	SALESMAN	7698	22-Feb-81	1250	500	30
7566	JONES	MANAGER	7839	02-Apr-81	2975		20
7654	MARTIN	SALESMAN	7698	28-Sep-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-May-81	2850		30
7782	CLARK	MANAGER	7839	09-Jun-81	2450		10
7788	SCOTT	ANALYST	7566	19-Apr-87	3000		20
7839	KING	PRESIDENT		17-Nov-81	5000		10
7844	TURNER	SALESMAN	7698	08-Sep-81	1500	0	30
7876	ADAMS	CLERK	7788	23-May-87	1100		20
7900	JAMES	CLERK	7698	03-Dec-81	950		30
7902	FORD	ANALYST	7566	03-Dec-81	3000		20
7934	MILLER	CLERK	7782	23-Jan-82	1300		10
7935	MILLER1	CLERK	7782	23-Jan-82	1300		

FK & Recursive Relationships

Foreign Keys Examples

Logical link between tables



Introduction to Database Administration

TOPIC 21

Foreign Keys Examples

Foreign Keys Examples

Foreign Keys Examples

FK is a part of composite key

Logical link between tables

Recursive Relationship

Foreign Keys Examples

FK is a part of Composite keys

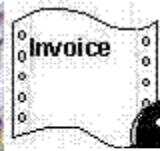
NOTE: PK column is underlined

ITEM (Item#, Item_Name, Department, Buyer)

ORDER_ITEM (OrderNumber, Item#, Quantity, Price, ExtendedPrice)

Where ORDER_ITEM.ITEM# must exist in ITEM.ITEM#

OrderNumber is an alternate name of



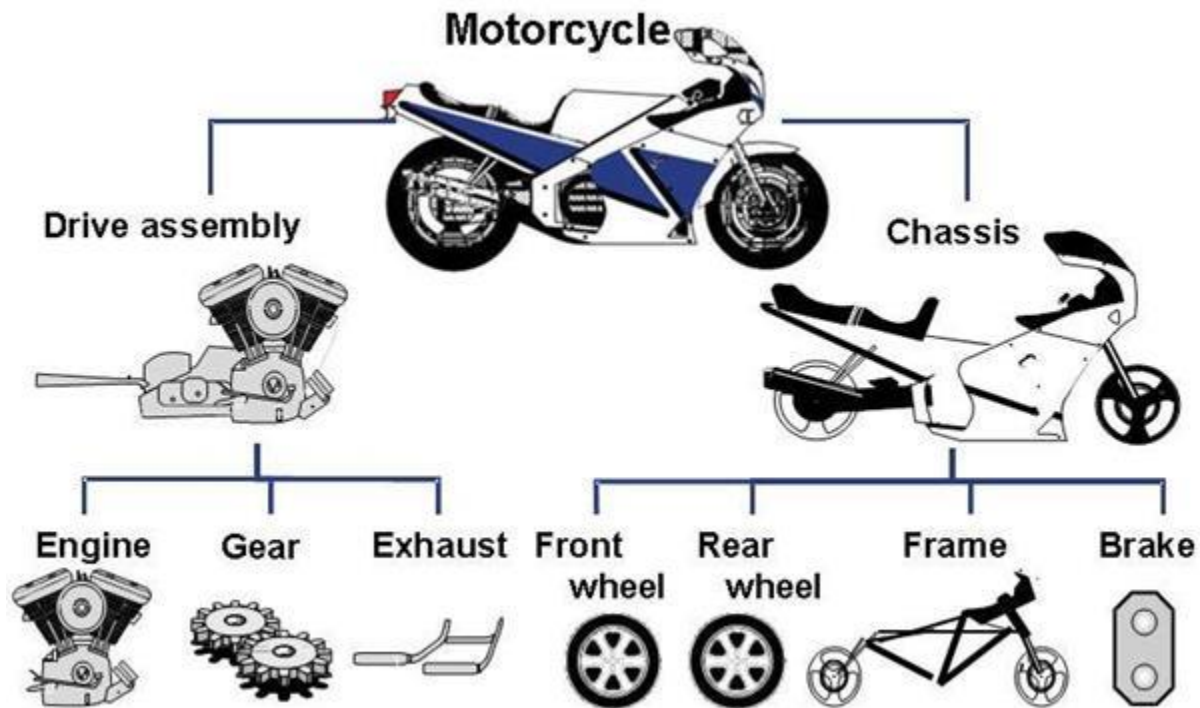
InvoiceNumber

Foreign Keys Examples

Referential Integrity Constraint

A **referential integrity constraint** is a statement that limits the values of the foreign key to those already existing as **primary key** values in the corresponding relation

Foreign Keys Examples



Recursive Relationship

Part (Part#, PName, Weight, MakeYear, Price, PType, RefPart#)

PType: Assembled product/ parts (e.g., Motorcycle) or single part

Introduction to Database Administration

TOPIC 22

Foreign Keys Examples

Foreign Keys Examples

Foreign Keys Examples

Integrity rules

Referential with cascade

Integrity Example

Foreign Keys Examples

Integrity Rules

Entity integrity

No two rows with the same primary key value

No null values in a primary key

Referential integrity

Foreign keys must match candidate key of source table

Foreign keys in some cases can be null

The database must not contain any unmatched foreign key values.

If B references A, A must exist.

Foreign Keys Examples

Referential with cascade

When we update a tuple from a table, say S, that is *referenced* by another table, say SP,

There are similar choices of *referential actions for update and delete*:

ON UPDATE CASCADE

ON UPDATE RESTRICT

There could be other choices besides these three. , e.g.,

ON UPDATE SET DEFAULT.

ON UPDATE SET NULL.

o	NAME	STATUS	CITY	so	SP	P#	QTY
P#	PNAME	COLOR	WEIGHT	CITY	S2	P1	300
P1	Nut	Red	12	London	S2	P2	400
P2	Bolt	Green	17	Paris	S2	P2	200
P3	Screw	Blue	17	Rome	S4	P2	200
P4	Screw	Red	14	London	S4	4	300
PM	Cam	ue	2	Paris	S4	as	400
P6	Cog	Red	If	London			

Integrity Example

81	Smith	20	London	S1	P 1	300
82	Jones	10	Paris	S 1	P2	200
83	Blake	30	Paris	SI	P3	400
84	Clark	20	London	S 1	P4	200
85	Adams	30	Athens	S 1	P5	100
					P6	100

s
Introduction to Database Administration

TOPIC 23

Composite Keys

Composite Keys

Composite Keys

Definition

Basic examples

Other Examples

Composite Keys

Definition of Composite Key

Combination of primary keys in a relation of selected attributes gives the concept of composite key.

Composite key is an extended form of a primary key.

All the characteristics of PK applies on Composite Keys more than one columns

Composite Keys

Basic Example of Composite Keys

During ATM Transaction, amounts can be drawn several times on one ATM Card.

We need follow the issues

How much amount will be withdrawn? When amount was drawn?

Which machine has been used? What are the types of transaction?

Card#, Amount, DrawDate, Machine#,

TransType are the attributes for transaction

Composite Keys

Other Examples of Composite Keys

Can we say Card# will be a primary key? NO Can DrawDate will be a key with Card#? NO

Then what to do? Add Surrogated key which is TransNo

ATMTransaction(Card#, TransNo, Amount, DrawDate, Machine#, TransType)

Date includes time in seconds as well.

Many transactions are managed on same date.

Introduction to Database Administration

TOPIC 24

Composite Keys Examples

Composite Keys Examples

Composite Key

Preparing Dataset for Composite Keys

Hospital Related Example

What about Excluding Composite Keys?

Composite Keys Examples

Preparing Dataset for Composite Key

Card(Card#, CardHolderName, ExpiryDate, IssueDate, Acc#)

ATMTransaction(Card#, TransNo, Amount, DrawDate, Machine#, TransType)

Do we need most closed related table? What are PK and FK?

Composite Keys Examples

Hospital Related Example

PatientVisit(PatID, VisitSNO, Vdate, DocID, Diagnosis,...)

LabTest(PatID, VisitSNO, TestID, Tdate, ...)

LabTest has **FKs** *PatID*, *VisitSNO*, referring to same corresponding composite keys in PatientVisit

Which lab table needs to be considered?

Composite Keys Examples

What about Excluding Composite Key?

Card(Card#, CardHolderName, ExpiryDate, IssueDate, Acc#)

ATMTransaction(ID, Card#, TransNo, Amount, DrawDate, Machine#, TransType)

Do we need most closed related table?

What is a drawback of using ID as Surrogated Key in ATMTransaction? What are PK and FK?

TOPIC 25

Composite Keys Examples

Composite Keys Examples

Composite Keys Examples

Course offering example

Preparing datasets

Composite Keys Examples

Course Offering Example

Let's us assume, there are 1000 number of courses in a table or list.

Students want to register in courses.

Can they register by looking at 1000 courses? NO

What to do?

Offer selected courses?

Composite Keys Examples

Course Offering Example

Offering course table

CrsOffer(SemID, CrsID, Sec, InstrName, B#, R#) What to do for keys, PK or Composite Key?

CrsOffer(SemID, CrsID, Sec, InstrName, B#, R#)

CrsReg(SemID, Roll#, CrsID, Sec, TotMarks, Grade)

Composite Keys Examples

Preparing Dataset for Composite Keys

CrsOffer(SemID, CrsID, Sec, InstrName, Building#, Room#)

CrsReg(SemID, Roll#, CrsID, Sec, TotMarks, Grade)

Do we need most closed related tables?

What are PK and FK?

Introduction to Database Administration

TOPIC 26

Composite Keys Examples

Composite Keys Examples

Composite Keys Examples

Meaningful learning-1

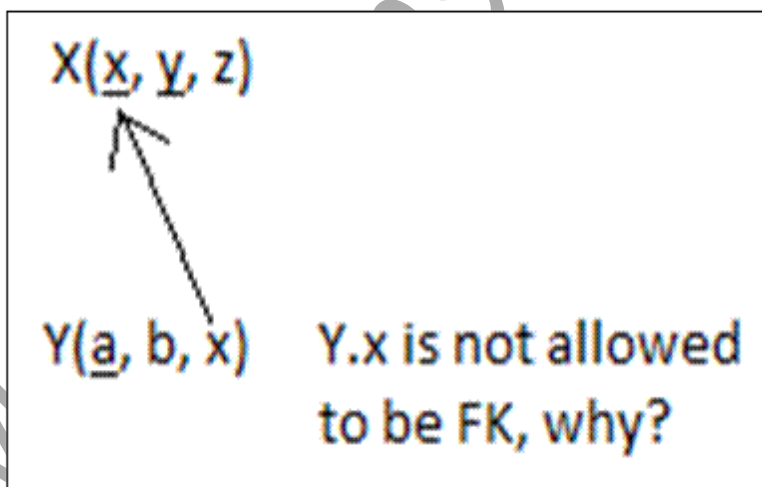
Meaningful learning-2

Wrong composite key

Indexing columns

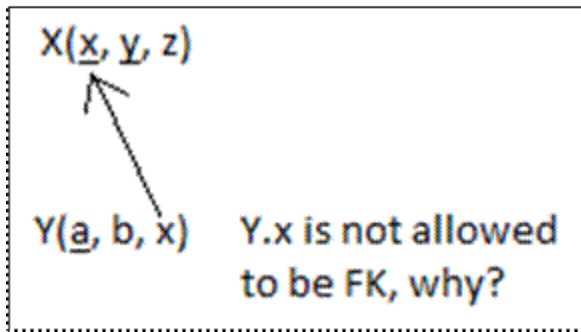
Composite Keys Examples

Meaningful learning-1



Composite Keys Examples

Meaningful learning-2



Analysis and Semantics
Evaluations

X					
	x		y		z
?	1		3		25
	1		6		32
	2		3		11
	3		1		55
	3		9		10
	3		3		12

Y		
a	b	x
121	0	2
122	3	1
123	5	3
124	6	

Semantics: Creating two tables with relevant data

Analysis: $Y.x=1$ cannot be mapped exactly with $X.x$ records 1, 3 or 1, 6

Evaluations: Default FK $Y.x$ cannot be created. Trigger can be created on $Y.x$ to make sure $Y.x$ will be a subset of $X.x$. Otherwise, data can be put in $Y.x$ which does not belong to or mapped to $X.x$.

Composite Keys Examples

Wrong composite key

**Do NOT make
composite Key**



Department

Department		LOC
DEPTNO	DNAME	
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON



PK

Without Indexing							Indexing columns	
PATIENT_ID	DEPEND_SN	PATIENT_VI	EMP_ID	DOCTOR_SN	VISIT_DATE	VISIT_NOTE		
0222082	0		13 6772356	35	12/27/2005 1:36:08 PM			
0257421	0		2 7033244	102	12/27/2005 3:23:10 PM			
1840190	6		7 6850269					
1976193	0		2 694108					
6901319	0		1 689156	0204531	0	15 6772356	28	10/3/2007 12:15:51 PM
6900626	0		1 694106	0204531	0	16 7073498	19	4/29/2008 1:40:11 PM
6911102	0		1 694106	0204531	0	17 6772356	37	4/29/2008 2:09:45 PM
0213919	0		1 702403	0204531	0	18 7024037	16	5/31/2008 9:20:18 AM
1731899	0		1 689156	0204531	0	19 7024037	4	8/5/2008 1:00:34 PM
6781521	8		2 691026	0204531	0	20 6772356	15	8/9/2008 8:36:01 AM
H500079	2		1 694106	0204531	0	21 7024037	4	8/11/2008 8:38:44 AM
0207842	0		1 677235	0204531	0	22 7073498	16	8/25/2008 1:01:39 PM
6781521	0		1 691026	0204531	0	23 6772356	7	10/26/2008 2:32:56 PM
1770371	0		1 701263	0204531	0	24 6772356	7	10/28/2008 8:56:20 AM
0227638	0		1 697135	0204532	0	25 7024037	39	11/10/2008 1:27:23 PM
1926081	1		2 694106	0204532	0	1 6850269	30	9/11/2005 7:17:21 PM
6751368	2		3 690055	0204532	0	3 7024037	50	10/5/2005 2:13:42 PM
				0204532	0	4 6850269	50	12/14/2005 2:46:16 PM
				0204532	0	5 6850269	59	1/24/2006 9:41:49 PM
				0204532	1	1 6850269	41	8/6/2005 2:53:29 PM
				0204532	1	2 7043508	31	9/6/2005 2:59:43 PM
				0204532	1	3 7043508	27	9/7/2005 3:25:09 PM
				0204532	1	4 6900551	12	9/8/2005 11:19:55 AM
				0204532	1	6 6971354	12	9/12/2005 11:12:52 AM
				0204532	1	7 6850269	57	1/24/2006 9:27:27 PM
				0204532	1	8 7024037	26	2/7/2006 2:54:09 PM
				0204532	2	1 7033244	3	10/22/2005 3:34:27 PM
				0204532	2	2 6971354	44	1/18/2006 2:54:51 PM
				0204573	0	1 7024037	35	1/21/2006 11:02:11 AM

Introduction to Database Administration

CS409 made by Mahjabeen
mahjabeen97869@gmail.com
 Contact# 03212711298

TOPIC 27

Additional or Other Constraints
 Additional or Other Constraints

Other constraints

What are additional or other constraints?

Null

Domain constraint

Additional or Other Constraints

What are additional or other constraints?

Other Constraints means additional constraints in addition to the standard constraints PK, FK, Composite Key, Surrogated key

For example, NOT NULL, CHECK, DEFAULT, UNIQUE etc.

Additional or Other Constraints

Null

Represents a value of an attribute that is currently unknown or is not applicable for this tuple, means nothing.

Cannot compute anything with the Null

Additional or Other Constraints

Domain constraint

Specifies that the value of attribute A must be an **atomic** value from the domain $DOM(A)$.
E.g., Width of column or Check on column.

Domain constraints must be Atomic means no two mobile numbers saved in one cell of a domain. Size of column must be appropriate.

S. RollNo	Name	Total:	20	20	10
30181-0599	Saifullah Dar		16	16	8
31181-0645	Muzammil Shakir		15	10	7.5

TOPIC 28

Additional Constraints using Tables

Additional Constraints using Tables

Other constraints

CHECK constraint

Unique & Default constraints

Composite unique constraints

Example of Trigger

Additional Constraints using Tables

CHECK constraints

The check clause in SQL permits domains to be restricted:

Use check clause to ensure that an hourly- wage domain allows only values greater than a specified value.

```
create domain hourly-wage numeric(5,2)
```

```
constraint value-test check(value >= 4.00)
```

The domain *hourly-wage* is declared to be a decimal number with 5 digits, 2 of which are after the decimal point

Additional Constraints using Tables

CHECK constraints

```
create table account
```

```
(branch-name Varchar(15),
```

```
acc-number Char(10) not null,
```

```
balance integer,
```

```
.....)
```

```
alter table account
```

```
add constraint ck1 check (bal > 0);
```

```
alter table emp
```

```
add constraint ck1_deptno check (deptno in (10,20,30));
```

Additional Constraints using Tables

UNIQUE & Default constraints

```
create table Dependent
```

```
(empno CHAR(6),
```

```
SNO Number(2),
```

DepName varchar2(25) NOT NULL, DOB Date,
 BloodGroup VARCHAR2(5),
 RelType CHAR(1) Default 'W');

alter table dependent
 add constraints pk_dep_empnoSNO primary key
 (empno, sno);

alter table dependent
 add constraints uniq_dependent unique (empno,
 DepName);

Additional Constraints using Tables

Composite UNIQUE constraints

SQL> select * from dependent;

EMPNO	SNO	DEPNAME	DOB	BGroup	RelType
123456	1	Ali	12-OCT-87	O+	S
123456	2	Samia	03-OCT-89	A+	D
123456	3	Ali	12-OCT-87	O+	S
123456	4	Sarah	12-OCT-70	B+	W
003477	1	Ahmed	11-NOV-89	B+	S

UNIQUE

alter table dependent
 add constraints uniq_dependent unique (empno, DepName);

HECK (Max_Participants>=Enrolled_Participants)

CHECK (Deptno>0)

CHECK (RelType IN ('S', 'D', 'W'))

Default 'Y'

Database Triggers

Additional Constraints using Tables

Example of Trigger

```

CREATE OR REPLACE TRIGGER display_salary_changes
BEFORE DELETE OR INSERT OR UPDATE ON EMP
FOR EACH ROW
WHEN (NEW.EMPNO > 0)
DECLARE
    sal_diff number;
BEGIN
    sal_diff := :NEW.sal - :OLD.sal;
    dbms_output.put_line('Old salary: ' || :OLD.sal);
    dbms_output.put_line('New salary: ' || :NEW.sal);
    dbms_output.put_line('Salary difference: ' ||
sal_diff);
END;
/

```

Introduction to Database Administration

TOPIC 29

Additional Constraints using DDL Alter SQL
Additional Constraints using DDL Alter SQL

Constraints with DDL Alter SQL

PK constraint on table

FK constraint on table

Composite unique & PK constraints

Additional Constraints using DDL Alter SQL

Primary key constraint on table

create table customer

(cid char(10),

Name varchar2(100) Not null, address varchar2(100),

city varchar2(20)

);

alter table customer

add constraints pk_cid primary key(cid);

Additional Constraints using DDL Alter SQL

Foreign key constraint on table

create table orders

(ordNo CHAR(5),

OrdType CHAR(1) not null, OrdDate Date,

TotalPrice Number(10,2), cid CHAR(10)

);

alter table orders

add constraints pk_ordNo primary key(ordNo);

alter table orders

add constraints fk_cid foreign key (cid) references customer(cid); --already created

Additional Constraints using DDL Alter SQL

Composite unique & PK constraints

create table Dependent (empno CHAR(6),

SNO Number(2),

DepName varchar2(25) NOT NULL, DOB Date,

BloodGroup VARCHAR2(5), RelType CHAR(1) Default 'W'); alter table dependent

add constraints pk_dep_empnoSNO primary key (empno, sno);

alter table dependent

add constraints uniq_dependent unique (empno, DepName);

Introduction to Database Administration

TOPIC 30

Additional Constraints using DDL Alter SQL

Additional Constraints using DDL Alter SQL

Constraints with DDL Alter SQL

Alter constraints ADD & Modify

Alter constraints using CHECK

Create table with multiple CHECK

Cascade and drop constraints

Additional Constraints using DDL Alter SQL

Alter constraints Add & Modify

Following constraints can be applied for an

existing EMP table:

```
alter table emp modify  
(sal constraint nt_null_emp_sal NOT NULL);
```

```
alter table emp  
add (constraint ck1_emp check (deptno>0));  
Additional Constraints using DDL Alter SQL
```

Alter constraints using CHECK

Following constraints can be applied for an existing EMP table:

```
alter table emp  
add (constraint ck2_emp check (sal + NVL(comm,0)<=6000));
```

--Null is allowed, if column is not null alter table emp

```
add constraint ck1_emp check (deptno in (10,20,30));
```

Additional Constraints using DDL Alter SQL

Create table with multiple CHECK

```
create table AirTraveller (TicketNo CHAR(10), FirstName VARCHAR2(20),  
FamilyName VARCHAR2(20) NOT NULL,  
Nationality VARCHAR2(25), BirthDate DATE,  
FlyCode Number,  
CONSTRAINTS ch_natval CHECK (Nationality IN ('Canadian','UK','Pakistan','Bahrain')),  
CONSTRAINTS ch_dob CHECK (BirthDate LIKE '[1-2][0-9][0-9][0-9]'),  
CONSTRAINTS ch_flycode CHECK (FlyCode between 3 and 9));
```

Additional Constraints using DDL Alter SQL

Cascade & drop constraints

Cascade drops the FK of child table EMP
emp.deptno references to dept.deptno

```
Alter table dept  
drop constraint pk_deptno cascade;  
Introduction to Database Administration
```

TOPIC 31

Keys & Constraints Revisions

Keys & Constraints Revisions

Constraints Revisions

Standard constraints are

Super Key

Candidate Key

Primary Key (PK)

Surrogated Key

Foreign Key (FK)

Composite Key

Additional constraints are

Not Null, Check, Unique, Default, DOM Constraints using triggers

Keys & Constraints Revisions

Super Key and Candidate Key

Super Key describes concepts to find unique records, can be null & indexed

{Roll#, FullName} gives unique when FullName can be duplicated

{CNIC}, {Barcode} are also a super keys

Candidate key gives fewer records in search.

Candidate keys can be used for searching purposes

{CNIC}, {Roll#} and {MedName} are

candidate keys

Keys & Constraints Revisions

Primary Key

Gives unique record with conditions NOT NULL, UNIQUE and INDEXED

Fixed length, ownership and not frequently changed, Descriptive/ well defined key

Roll#, Empno, Item#, PatientID etc. 2019004, 16K-1005 etc.

Avoid PKs Price, Quantity, Rates

Keys & Constraints Revisions

Surrogated Key

Just like a PK but it is artificial key. Decided by the designer from set of attributes and meaningless to the user.

SerialNo or Card# with TransNo

Try to avoid Column with datatype DATE

Keys & Constraints Revisions

Foreign Key

Referencing key and also called referential integrity constraint

Dept(DNO, Dname, Loc) Emp(Empno, Ename, Sal, DOB, **DNO**)

Emp.DNO is a FK references to Dept.DNO

FK is also Recursive Key!

Keys & Constraints Revisions

Composite Key

Collection of PKs in a relation

PatientVisit(PatID, **VisitSNO**, Vdate,
DocID, Diagnosis,...)

LabTest(PatID, VisitSNO, **TestID**, Tdate,
...)

Columns with red text are surrogated keys

Keys & Constraints Revisions

Additional Constraints

NOT NULL DOMAIN CHECK UNIQUE

End

DEFAULT and TRIGGER

Introduction to Database Administration

TOPIC 32

Script of Referential Integrity Constraints SQL

Script of Referential Integrity Constraints SQL

Script of Referential Integrity

Primary key with table

Primary & Foreign Key with table

Employee Prepares Order

Item Table

Sold Table & FKs

Script of Referential Integrity Constraints SQL

Primary key with Table

Follow up all DDL in a given order of slides create table Customer
(custID CHAR(10),
CustName VARCHAR2(25) NOT NULL, Address VARCHAR2(50),
City VARCHAR2(50),
TelNo VARCHAR2(15),
Email VARCHAR2(50));

Alter table Customer
add constraint PK_Custid primary key
(CustID);
Script of Referential Integrity Constraints SQL

Primary & Foreign keys Create table Orders (InvoiceNo Varchar2(12), InvType CHAR(1),
InvDate Date NOT NULL, TotalPrice Number(6,2), CustID CHAR(8),
SalesEmpID CHAR(5));

Alter table Orders
add constraint PK_InvNo primary key (InvoiceNO);
Alter table Orders
add constraint FK_Cust_CustID foreign key (custID) references Customer(CustID);
Script of Referential Integrity Constraints SQL

Employee Prepares Order

create table Employee (EmpID CHAR(3),
EmpName VARCHAR2(25) NOT NULL, Address VARCHAR2(50),
City VARCHAR2(50),
TelNo VARCHAR2(15) UNIQUE,
Email VARCHAR2(50));
Alter table Employee
add constraint PK_EmpID primary key (EmpID);
Alter table Orders
add constraint FK_Emp_EmpID foreign key (SalesEmpID) references Employee(EmpID);
Script of Referential Integrity Constraints SQL

Item Table

Create table Item (ItemID CHAR(5),
ItemName VARCHAR2(50) NOT NULL,
PPrice NUMBER(10,2),
TotQty NUMBER(5),
Status CHAR(1)
);

Alter table Item
add constraint PK_ItemID primary key (ItemID);
Script of Referential Integrity Constraints SQL

Sold Table & FKs Create table Sold (ItemID CHAR(5),
InvoiceNo CHAR(12),
SPrice NUMBER(10,2) Not Null,
SQty NUMBER(10,2) Not Null
);
Alter table Sold
add constraint PK_Sold primary key (ItemID, InvoiceNo); --Composite Key
Script of Referential Integrity Constraints SQL
Sold Table & FKs

Alter table Sold
add constraint FK_Sold_ItemID
foreign key (ItemID) references Item(ItemID);

Alter table Sold
add constraint FK_Sold_InvoiceNo
foreign key (InvoiceNo) references Invoice(InvoiceNo);

End
Introduction to Database Administration

TOPIC 33

Fundamentals of ER Model
Fundamentals of ER Model

Fundamentals of Entity Relationship (ER) Model

Semantic Modeling
Chen ER Model
System Development & DBA
Database Development
Conceptual Model
Conceptual Design
Fundamentals of ER Model

Semantic Modeling
A class of data models
Conveys semantic meaning
Implements databases more intelligently

Support more sophisticated user interfaces (SQL queries, forms & reports)

Research started in 1970s and early 1980s

Other names are data modeling, E/R modeling, E-modeling

Fundamentals of ER Model

Chen ER Model

Chen in 1976 introduced the E/R model

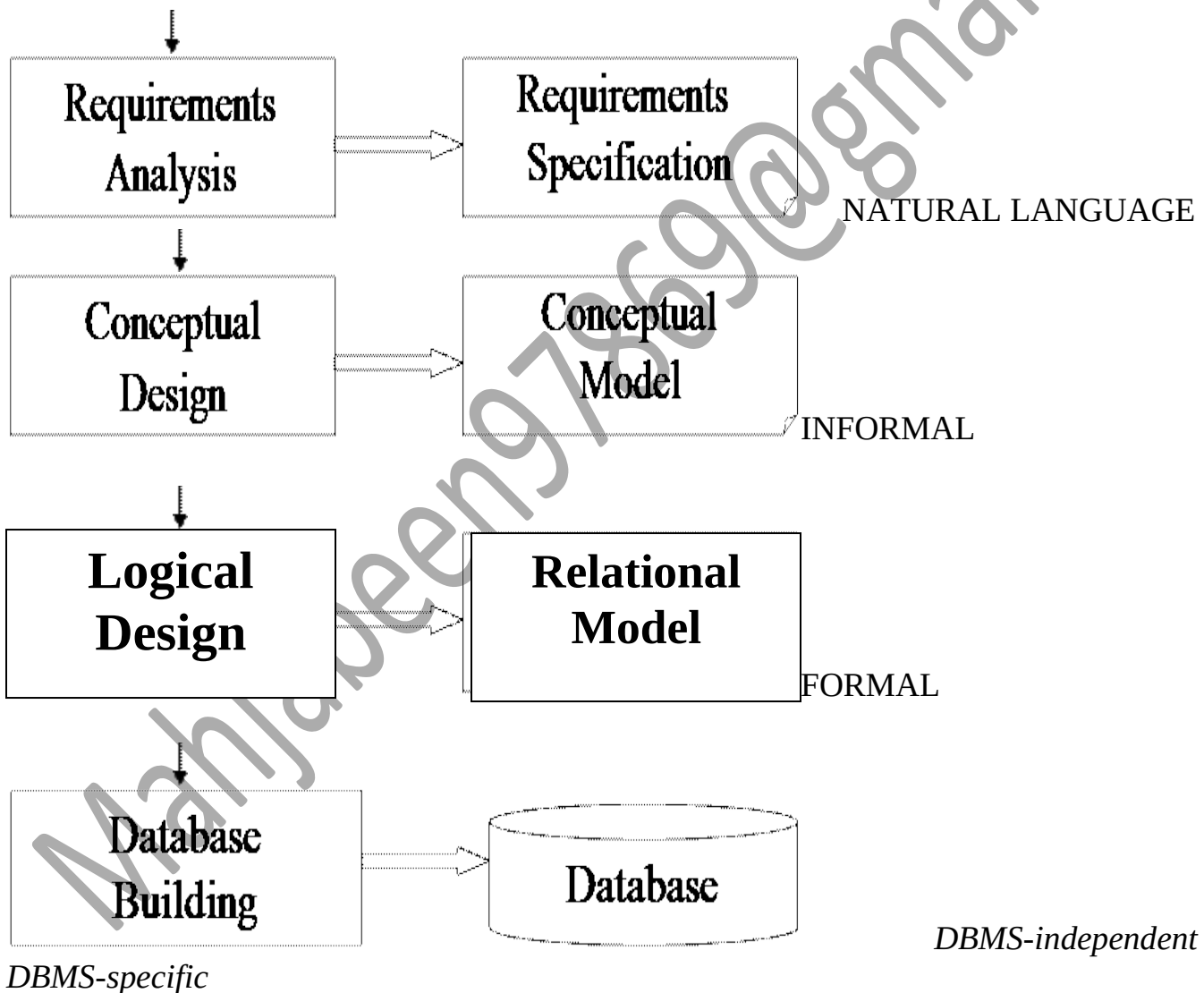
The E/R model allows us to sketch the design of a database informally.

Designs are pictures called *entity-relationship diagrams*.

Fairly mechanical ways to convert E/R diagrams to real implementations like relational databases exist.

Fundamentals of ER Model

System Development & DBA



Operation, Maintenance, & Tuning



Evolving Database

Physical/ DBA

Fundamentals of ER Model

Database Development

☐ **Requirements Analysis**

☐ Collect and Analyze the requirements of the users, e.g., Functional Specification, Prototyping.

☐ **Conceptual Design**

☐ Design a conceptual model (schema), e.g., ER model, ER Mapping.

☐ **Logical Design**

☐ Translate the ER model into a relational model (schema), DDL Scripts.

☐ Normalization

☐ **Database Building**

☐ Build the database and write application programs for specific DBMS & S/W Appls/ Tools.

☐ **Operation, Maintenance & Tuning**

☐ Use/ Training, Installations, maintain and “tune” the database.

Fundamentals of ER Model

Conceptual Model

Highest conceptual grouping of ideas

Data tends to naturally cluster with data from the same or similar categories relevant to the organization

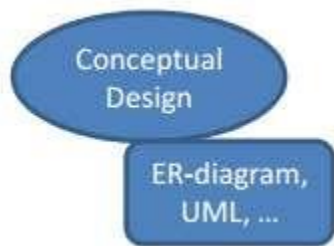
The major relationships between subjects have been defined



Least amount of detail

Fundamentals of ER Model

Conceptual Design



Entity-Relationship (E R) Modeling

Entities - "things" in the real world

Car

Account

Product

E.g. Car, Account, Product

Attributes – property of an entity, entity type, or

Car

Color

relationship type

E.g. color of a car, balance of an account, price of a product

Relationships – between entities there can be relationships, which also can have attributes



E.g. Person owns Car

Introduction to Database Administration

TOPIC 34

ER Diagram & Attributes Display

ER Diagram & Attributes Display

Attributes Display

For Logical Design

Symbols/ Notations

Motivational Example

Entity or Entity Type

Simple or Single Attributes

Composite Attributes

Multi valued Attributes

Derived or Stored Attributes

Complex Attributes

ER Diagram & Attributes Display

For Logical Design

Translate an ER schema into a relational schema.

Entity

Relationship

Attribute

Subtype

Normalization

To ensure you have a “good” design (Testing of your model).

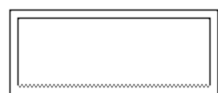
ER Diagram & Attributes Display

Symbols/ Notations

Symbol Meaning



ENTITY TYPE



WEAK ENTITY TYPE



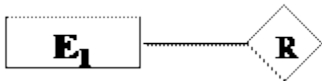
RELATIONSHIP TYPE



IDENTIFYING RELATIONSHIP TYPE ATTRIBUTE



KEY ATTRIBUTE MULTIVALUED ATTRIBUTE COMPOSITE ATTRIBUTE



N



$R(\underline{\text{min,max}})$

DERIVED ATTRIBUTE



TOTAL PARTICIPATION OF E_2 IN R



CARDINALITY RATIO 1:N FOR $E_1:E_2$ IN R

E

STRUCTURAL CONSTRAINT (min, max) ON PARTICIPATION OF E IN R
ER Diagram & Attributes Display

Motivational Example

Consider building a course management system (CMS):

Students
Courses
Professors



Entities

Who takes what?
Who teaches what?



Relationships

What about Sales Items System?
ER Diagram & Attributes Display

Entity or Entity Type

An entity is an object or a concept that is identified by the enterprise as having an independent existence.

There are two types of entities:

Physical entities

Example: car, house, employee, part, supplier, building

Conceptual entities

Example: job, university course, sale, work experience, CourseReg

ER Diagram & Attributes Display

Simple or Single Attributes

A simple attribute:

Consists of a single component with an independent existence.

Example: The Sex or Gender attribute of an EMPLOYEE entity

Example: Blood Group of a student

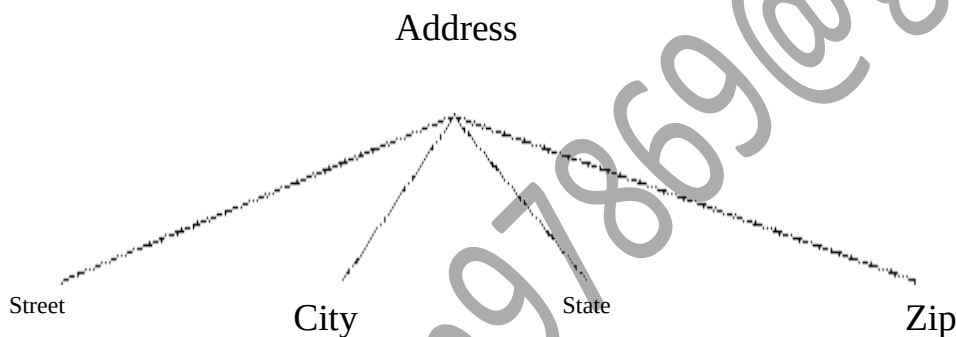
ER Diagram & Attributes Display

Composite Attributes

A composite attribute:

Consists of multiple components each with an independent existence.

Example: The Address attribute of an EMPLOYEE entity can be divided into Street, City, State, Zip.



ER Diagram & Attributes Display

Multivalued Attributes

An attribute that holds more than one value for a single entity.

Example: The Color attribute of a CAR entity, more than college degrees taken by students

Example: Multiple Phone numbers

ER Diagram & Attributes Display

Derived & Stored Attributes

Is an attribute that represents a value that is derived from the value of a related attribute or set of attributes not necessarily in the same entity.

Example 1: The value of the Age attribute of the EMPLOYEE entity can be derived from the today's date and the value of the employee BirthDate.

Example 2: The NumberOfEmployees attribute of a DEPARTMENT entity can be derived from the EMPLOYEE table by counting the number of employees who work in that department.

ER Diagram & Attributes Display

Complex Attributes

Is an attribute that is a nested combination of composite and multi-valued attributes in an arbitrary way.

Example: The complex attribute AddressPhone.

```
{AddressPhone({Phone(AreaCode,  
PhoneNumber}),  
Address(StreetAddress(Number, Name,  
HouseNo), City, State, Zip)))}
```

Note

Comma is used for separating the components

{ } Represents multi-valued attributes

Introduction to Database Administration

TOPIC 35

Fundamental Notations of ER Diagram

Fundamental Notations of ER Diagram

Attributes Display

Symbol/ Notations

Entity Employee

Entity Department

Entity Project

Entities Department & Employee

Fundamental Notations of ER Diagram

Symbols/ Notations

Symbol Meaning



ENTITY TYPE



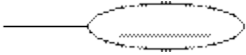
WEAK ENTITY TYPE



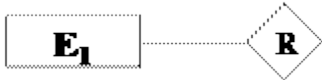
RELATIONSHIP TYPE



IDENTIFYING RELATIONSHIP TYPE ATTRIBUTE



KEY ATTRIBUTE MULTIVALUED ATTRIBUTE COMPOSITE ATTRIBUTE

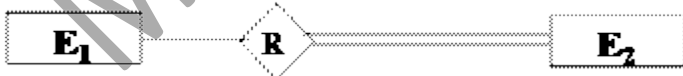


N

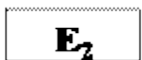


$R(\underline{\min, \max})$

DERIVED ATTRIBUTE



TOTAL PARTICIPATION OF E_2 IN R



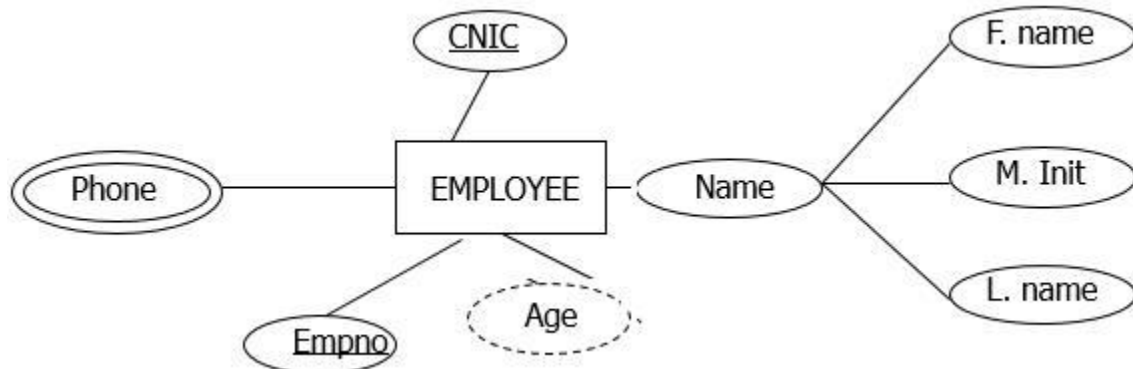
CARDINALITY RATIO 1:N FOR $E_1:E_2$ IN R

E
STRUCTURAL CONSTRAINT (min, max) ON PARTICIPATION OF E IN R
Fundamental Notations of ER Diagram

Entity Employee

Underlined are Key Identifiers

There is no PK and FK are shown in ER Diagram



Fundamental Notations of ER Diagram

Entity Department

Underlined are Key Identifiers

There is no PK and FK are shown in ER Diagram

Double oval represents multi-valued



Fundamental Notations of ER Diagram

Entity Project

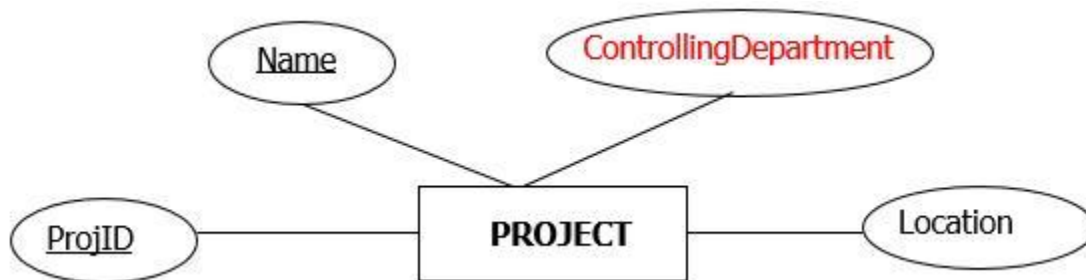
An entity type PROJECT has attributes:

ProjID (key identifier)

Name (key identifier)

DNumber (key identifier)

Location
ControllingDepartment
Budget
Start Date



End Dates
Fundamental Notations of ER Diagram
Entities Department & Employee

Consider Employees are working in departments:

Department
Employee
Manager/ Boss??

Entities



Employees are belong to Departments
Some employees are managers of department
Employee may work in more than one dept
Some are managers of other employees
Introduction to Database Administration

Relationships



TOPIC 36

ER Diagram with Relationship one to many

ER Diagram with Relationship one to many

Attributes Display

Example-1 (1-M)

Example-2 (1-M)

Example-3 (1-M)

Example-4 (1-1)

Exercise

ER Diagram with Relationship one to many

Example-1 (one-many)

Any Employee has his/ her own department, where works in

One department has many employees or

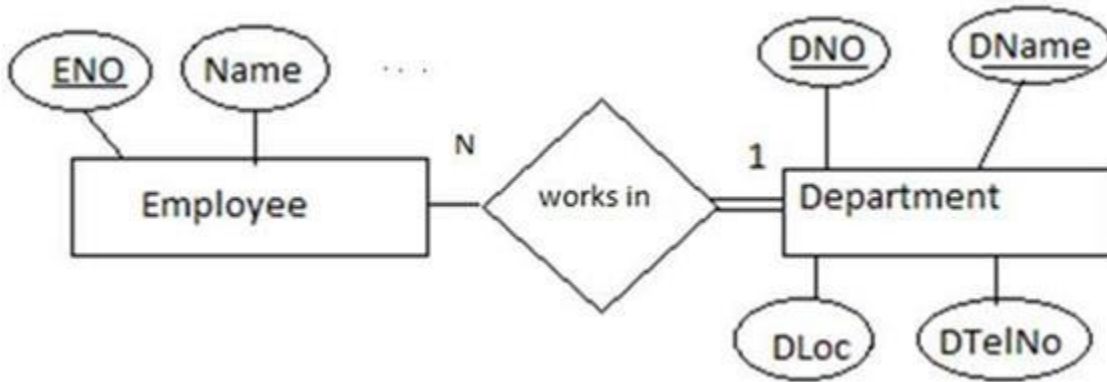
Employee belongs to department



ER Diagram with Relationship one to many

Example-2 (one-many)

Any Employee has his/ her own department, where works in
Any one department has at least **ONE** or many employees
Employee belongs to department



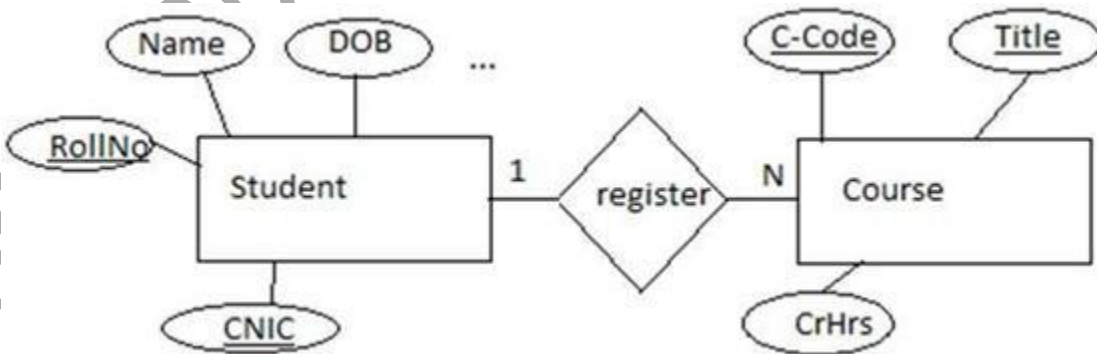
ER Diagram with Relationship one to many

Example-3 (one-many)

Below diagram is NOT correct.

Any one student can takes or registers
many courses

But any one course is taken by many
students



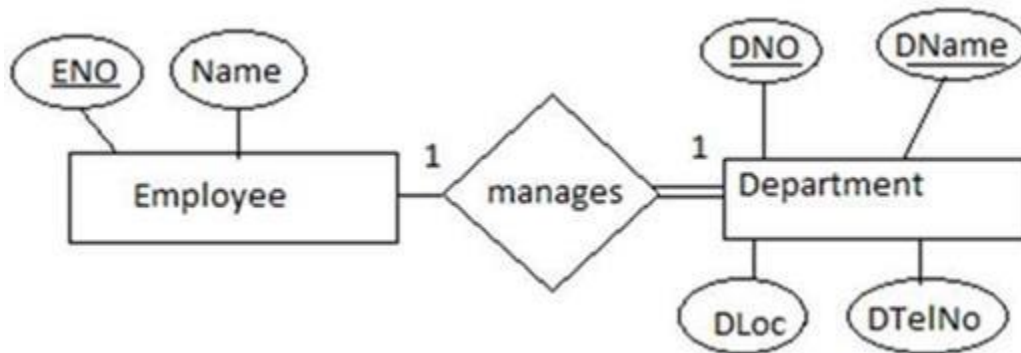
ER Diagram with Relationship one to many

Example-4 (one-one)

Some of employees will be a manager of departments.

Any one employee will be a manager of any one department but at the same time cannot be a manager of multiple departments

Any one department has one manager



as an employee.

ER Diagram with Relationship one to many

Exercise

Combining ER Diagrams of both one to one and one to many Entities are Department & Employee

Introduction to Database Administration

TOPIC 37

ER Diagram with Relationship many to many

ER Diagram with Relationship many to many

Attributes Display

Example-1 (M-N)

Example-2 (M-N)

Example-3 (M-N)

Example-4 (M-N)

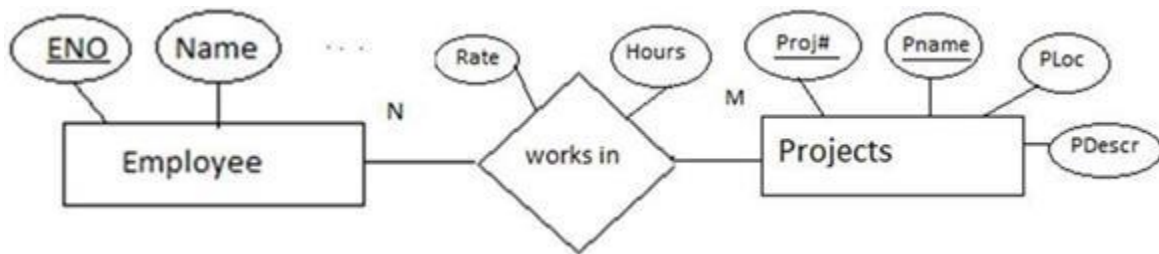
Exercise

ER Diagram with Relationship many to many

Example-1 (many-many)

Any Employee has his/ her own department, where works in as full time employee.

Additionally, Employee works in a Project with number of **hours** and **rate** per hour

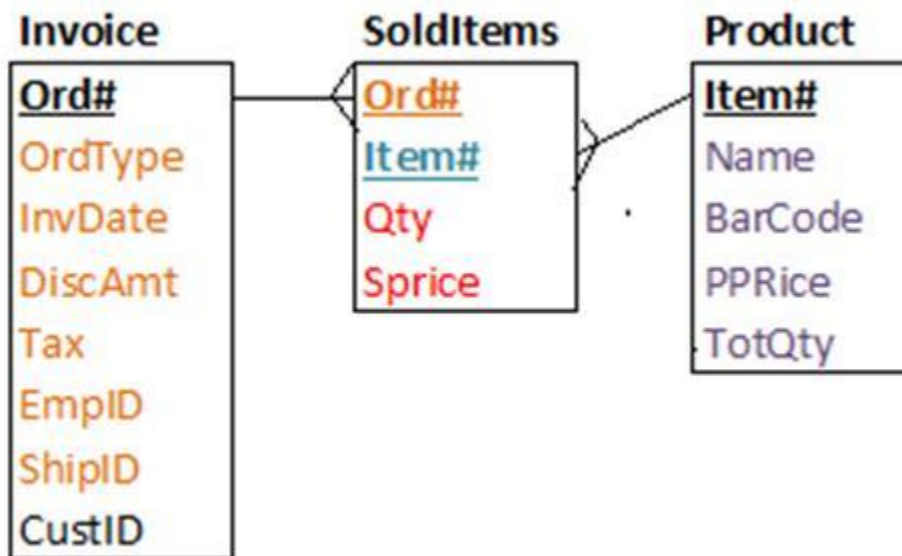


ER Diagram with Relationship many to many

Example-2 (many-many)

Customer buys items with invoice.

Any one invoice contains many products/ items and same product is sold on many invoices.



ER Diagram with Relationship many to many

Example-3 (many-many)

Every course is offered with book as its recommended and reference books.

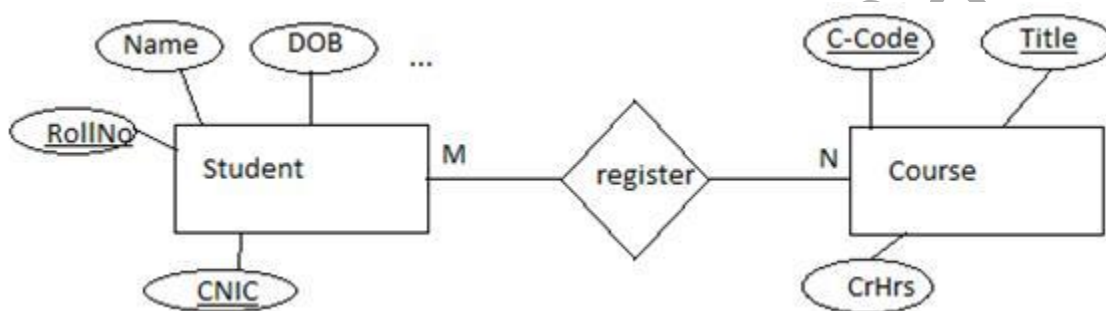
Any book can referred in many courses.



ER Diagram with Relationship many to many

Example-4 (many-many)

Below diagram is no longer be valid if student re-enrolled in course again.
Enrollment requires semester as well.



ER Diagram with Relationship many to many

Exercise

Exercise-4 needs to be drawn with correctness, using semester entity after course offering. Such diagram is called ternary ER relationship.

Introduction to Database Administration

TOPIC 38

ER Diagram with Identifying & Weakness Relationship

ER Diagram with Identifying & Weakness Relationship

Attributes Display

Identifying & Weakness

Identifying & Identified entities

Example-1

Example-2

ER Diagram with Identifying & Weakness Relationship

Identifying and weakness

An entity whose existence depends on some other entity type is called a **weak** entity type. Otherwise it is called a **strong entity** type.

There is a chance of records' duplications on non-key attributes in weak entity types.

Child entity depends on other parent entity. Parent entity is identifying the child entity.

Weakness can be reduced only.

ER Diagram with Identifying & Weakness Relationship

Identifying and identified entities

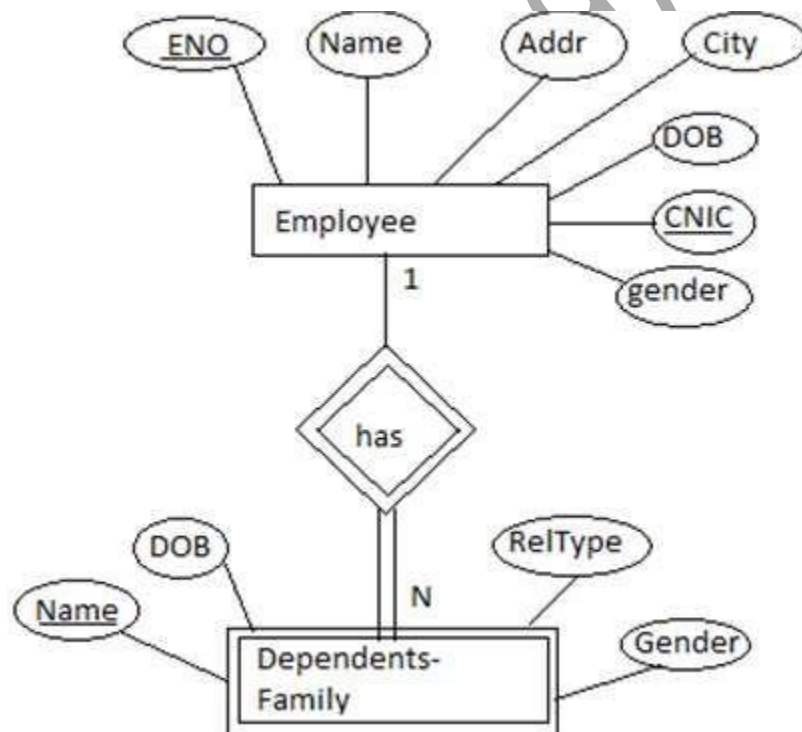
All ID-Dependent (identified) entities **MAY BE** considered weak.

Examples are

Employee & Family Members ATM Card & Transactions, Patient and Patient visits, Patient & Issue Medicines, Loan and its Installments

ER Diagram with Identifying & Weakness Relationship

Example-1



EMPLOYEE(ENO, Name, Gender, DOB, Addr, City, CNIC)