

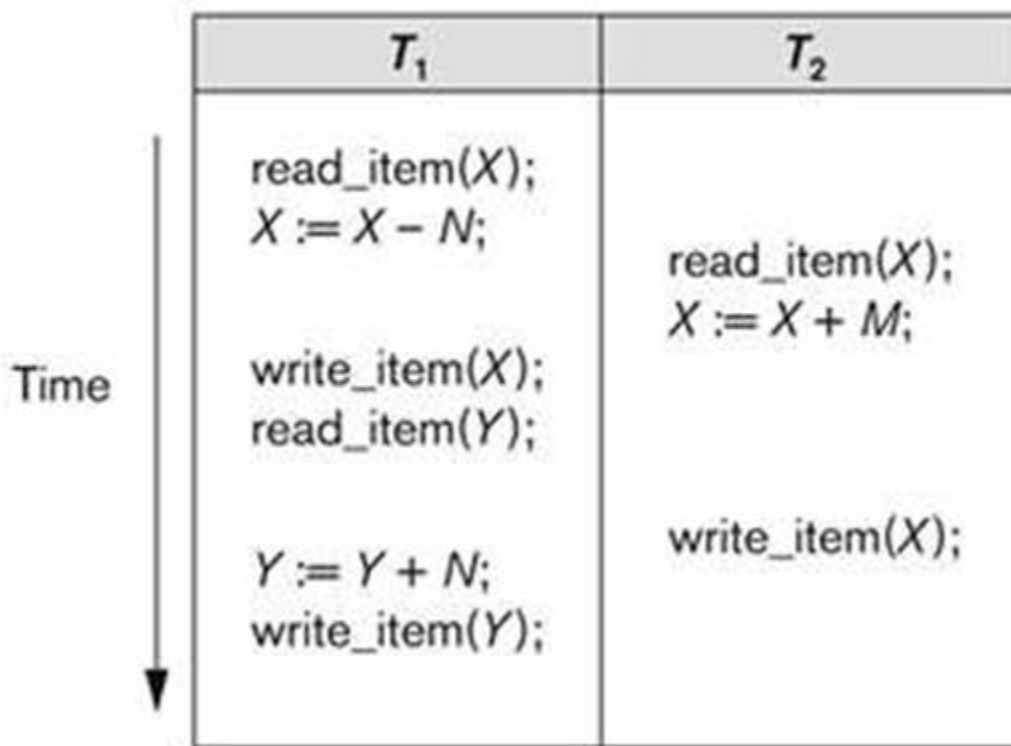
Serial Schedules Example-2

Schedule with Serial Schedules Example-2

Example-2

Serial Schedules Example-2

Example-2 (Schedule S')



Assume $X=90$, $Y=90$, $N=3$, $M=2$

Not Correct Values

Introduction to Database Administration

TOPIC 239

Serial Schedules Example-3

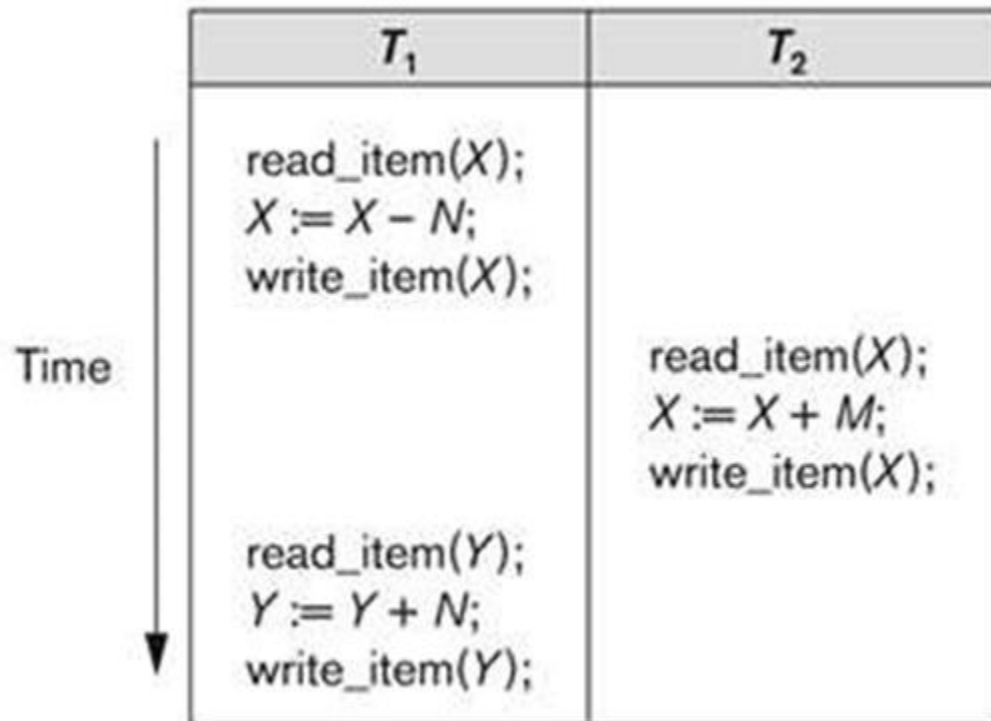
Serial Schedules Example-3

Schedule with Serial Schedules Example-3

Example-3

Serial Schedules Example-3

Example-3 (Schedule S'')



Assume $X=90$, $Y=90$, $N=3$, $M=2$

Produces same result as serial schedule

Introduction to Database Administration

TOPIC 240

Conflict Serializability Example-1

Conflict Serializability Example-1

Schedule with Conflict Serializability Example-1

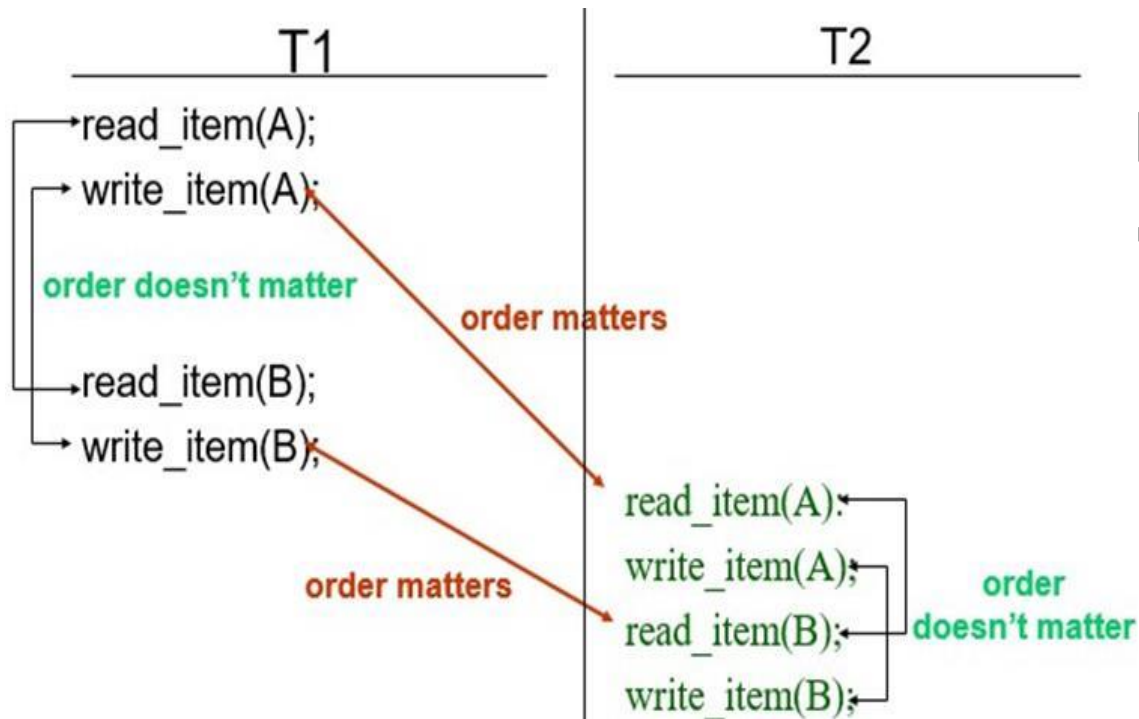
Example-1 (S1)

Example-1 (S1')

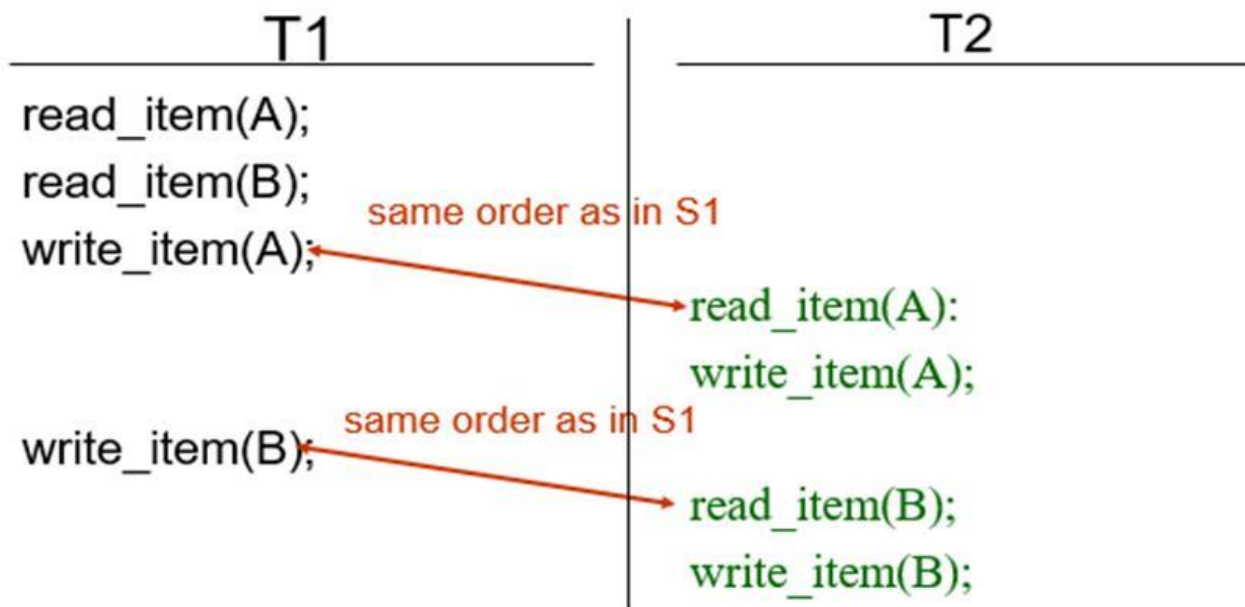
Example-1 (Schedules A, B)

Conflict Serializability Example-1

Example-1 (Schedule S)



Conflict Serializability Example-1



S1 and S1' are conflict equivalent
(S1' produces the same result as S1)

Example-1 (Schedule S1')

Conflict Serializability Example-1

Example-1 (Schedules A, B)

Schedule A		Schedule B	
T1	T2	T1	T2
read_item(X);		read_item(X);	
X:=X-N;		X:=X-N;	
write_item(X);		write_item(X);	
read_item(Y);			read_item(X);
Y:=Y+N;			X:=X+M;
write_item(Y);			write_item(X);
	read_item(X);	read_item(Y);	
	X:=X+M;	Y:=Y+N;	
	write_item(X);	write_item(Y);	

B is conflict equivalent to A $\Rightarrow$ B is serializable

TOPIC 241

Conflict Serializability Example-2 (Graph Test)

Conflict Serializability Example-2 (Graph Test)

Conflict Serializability Example-2 (Graph Test)

Test for Serializability

Example-2 (Graph Test)

Conflict Serializability Example-2 (Graph Test)

Test for Serializability

Construct a directed graph, *precedence graph*, $G = (V, E)$

V: set of all transactions participating in schedule

E: set of edges $T_i \rightarrow T_j$ for which one of the following holds:

T_i executes a write_item(X) before

T_j executes read_item(X)

T_i executes a read_item(X) before

T_j executes write_item(X)

T_i executes a write_item(X) before

T_j executes write_item(X)

Conflict Serializability Example-2 (Graph Test)

Test for Serializability

An edge $T_i \rightarrow T_j$ means that in any serial schedule equivalent to S, T_i must come before T_j

If G has a cycle, then S is not conflict serializable

If not, use topological sort to obtain serializable schedule (linear order consistent with precedence order of graph)

To check consistency if graph is acyclic otherwise schedule has deadlock means inconsistent

Conflict Serializability Example-2 (Graph Test)

Example-2 (Graph Test)

T1	T2	T3
read_item(X); write_item(X);		read_item(Y); read_item(Z);
read_item(Y); write_item(Y);	read_item(Z);	write_item(Y); write_item(Z);
	read_item(Y); write_item(Y); read_item(X); write_item(X);	

Introduction to Database Administration

TOPIC 242

Apply Locking
Apply Locking

Apply Locking

Database Locks
Locks in Oracle SQL
Locks Update
Apply Locking

Database Locks

When two sessions or users of database try to update or delete the same data in a table, then there will be a concurrent update problem. In order to avoid this problem, database locks the data for the first user and allows him to update/delete the data.

Once he is done with his update/delete, he COMMITs or ROLLBACK the transaction, which will release the lock on the data. When lock on the data is released, another user can lock it for his changes.

Apply Locking

Database Locks

An Object is locked before it performs any operation

Two types of operations, two types of locks

Shared & Exclusive Locks

Apply Locking

Locks in Oracle SQL

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
7369	SMITH	CLERK	7902	17-DEC-80	900
7499	ALLEN	SALESMAN	7698	20-FEB-81	1650
7521	WARD	SALESMAN	7698	22-FEB-81	1250
7566	JONES	MANAGER	7839	02-APR-81	2975
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250
7698	BLAKE	MANAGER	7839	01-MAY-81	2850
7782	CLARK	MANAGER	7839	09-JUN-81	2450
7788	SCOTT	ANALYST	7566	09-DEC-82	3000
7839	KING	PRESIDENT		17-NOV-81	5000
7844	TURNER	SALESMAN	7698	08-SEP-81	1500
7876	ADAMS	CLERK	7788	12-JAN-83	1100
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
7900	JAMES	CLERK	7698	03-DEC-81	950
7902	FORD	ANALYST	7566	03-DEC-81	3000
7934	MILLER	CLERK	7782	23-JAN-82	1300

14 rows selected.

```
SQL> update emp
2 set sal=950
3 where ename='SMITH';
```

1 row updated.

SQL>

Rollback complete.

```
SQL> select * from emp;
```

LOCKED by other session

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
7369	SMITH	CLERK	7902	17-DEC-80	900
7499	ALLEN	SALESMAN	7698	20-FEB-81	1650
7521	WARD	SALESMAN	7698	22-FEB-81	1250
7566	JONES	MANAGER	7839	02-APR-81	2975
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250
7698	BLAKE	MANAGER	7839	01-MAY-81	2850
7782	CLARK	MANAGER	7839	09-JUN-81	2450
7788	SCOTT	ANALYST	7566	09-DEC-82	3000
7839	KING	PRESIDENT		17-NOV-81	5000
7844	TURNER	SALESMAN	7698	08-SEP-81	1500
7876	ADAMS	CLERK	7788	12-JAN-83	1100

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
7900	JAMES	CLERK	7698	03-DEC-81	950
7902	FORD	ANALYST	7566	03-DEC-81	3000
7934	MILLER	CLERK	7782	23-JAN-82	1300

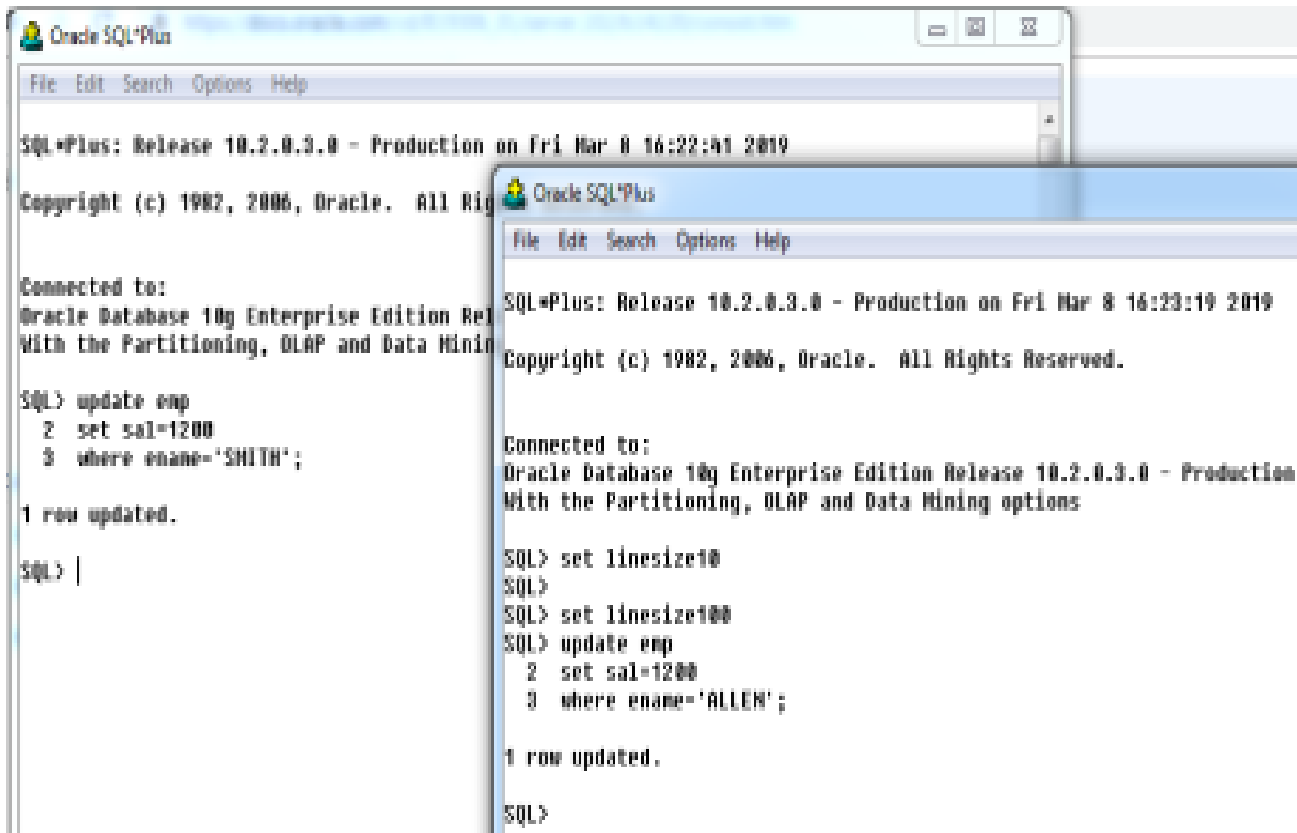
14 rows selected.

SQL>

```
SQL> update emp
2 set sal=1100
3 where ename='SMITH';
```

Record level lock/ Row Lock – same record

Locks Update



The image shows two overlapping Oracle SQL*Plus windows. The background window displays the following text:

```
Oracle SQL*Plus
File Edit Search Options Help

SQL*Plus: Release 10.2.0.3.0 - Production on Fri Mar 8 16:22:41 2019

Copyright (c) 1982, 2006, Oracle. All Rights Reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> update emp
  2 set sal=1200
  3 where ename='SMITH';

1 row updated.

SQL> |
```

The foreground window displays the following text:

```
Oracle SQL*Plus
File Edit Search Options Help

SQL*Plus: Release 10.2.0.3.0 - Production on Fri Mar 8 16:23:19 2019

Copyright (c) 1982, 2006, Oracle. All Rights Reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> set linesize10
SQL>
SQL> set linesize100
SQL> update emp
  2 set sal=1200
  3 where ename='ALLEN';

1 row updated.

SQL>
```

Introduction to Database Administration

TOPIC 243

Deadlocks

Deadlocks

Deadlocks

Database Deadlocks

Deadlocks - Two Transactions

Applying Locks at Table Level

Deadlock - Example

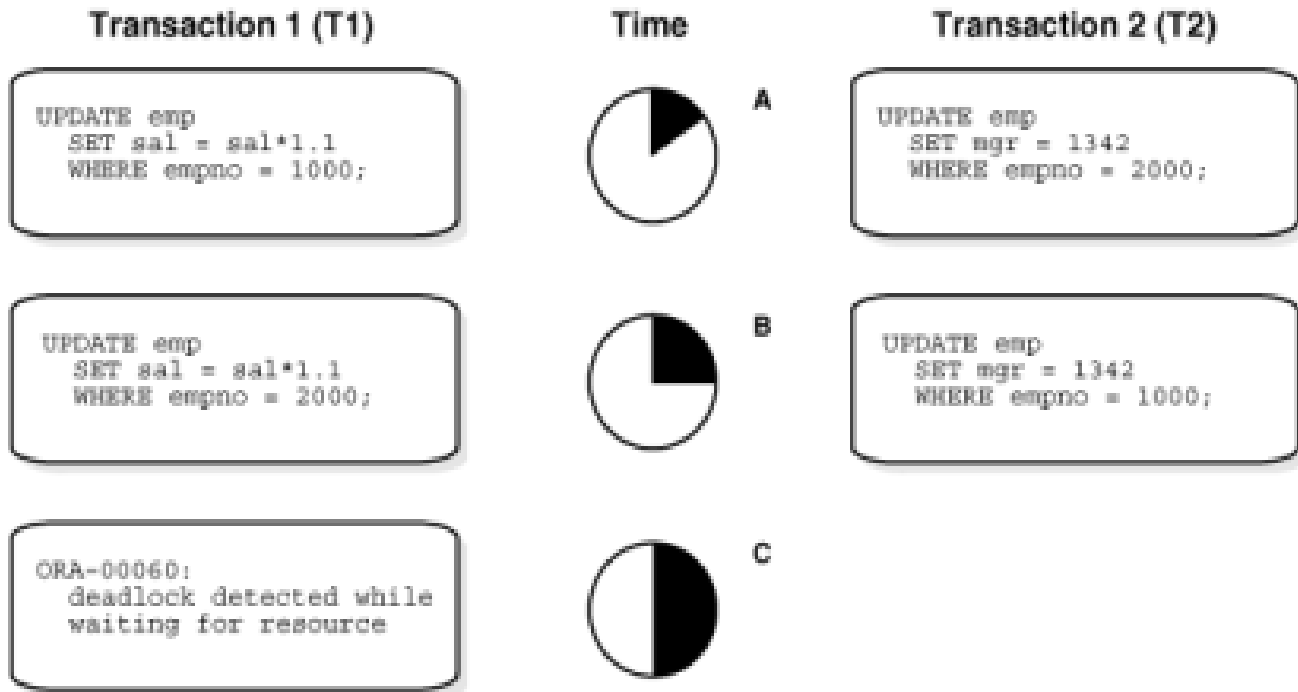
Deadlocks

Database Deadlocks

In a database, a deadlock is a situation in which two or more transactions are waiting for one another to give up locks.

For example, Transaction A might hold a lock on some rows in the Accounts table and needs to update some rows in the Orders table to finish.

Deadlocks – Two Transactions



Applying Locks at Table Level

```

--user scott
Lock table dept IN EXCLUSIVE MODE;

--user can use rollback to unlock this table

--user sys
select object_id, session_id, oracle_username, locked_mode
from V$Locked_Object;

OBJECT_ID SESSION_ID ORACLE_USERNAME LOCKED_MODE
-----
53768 140 SCOTT 6 (Locked applied)
53768 148 SCOTT 0 (Locked in other session)

--in session 148
SQL> update dept
2 set loc='BOSTON1'
3 where deptno=40;

-----LOCKED-----

How to put in share mode?
Lock table dept in share mode;

```

Deadlock - Example

Tiw	T1	T2
t1	XLek(x)	
t2	..	XLek(y)
t3	Request y	
t4	wait	Request x
t5	wait	wait
t6	wait	wait

TOPIC 244

DDL Schema Script for Project

DDL Schema Script for Project

DDL Schema Script for Project

Plan the Script for Schema

Sample Script

DDL Schema Script for Project

Plan the Script for Schema

ANSI standard script of SQL consists of Create and Alter Statements

Standard & Additional Constraints

Tables with Tablespace Storage

Database Security

Shared Logical Objects such as Views, Synonyms

Logical & Physical Storage DML - SQL

DDL Schema Script for Project

```

Drop table employee;
Drop table orders;
Drop table customer;
Drop table Item;

create table Customer
(custID          CHAR(10),
CustName        VARCHAR2(25) NOT NULL,
Address         VARCHAR2(50),
City            VARCHAR2(50),
TelNo           VARCHAR2(15),
Email           VARCHAR2(50) UNIQUE)
tablespace TS_Sales;

Alter table Customer
add constraint PK_Custid primary key (CustID);

Create table Orders
(InvoiceNo       CHAR(12),
InvType         CHAR(1),
InvDate         Date NOT NULL,
TotalPrice      Number(6,2),
SalesEmpID      CHAR(5),
CustID          CHAR(8))
tablespace TS_Sales;

Alter table Orders
add constraint PK_InvNo primary key (InvoiceNO)
USING INDEX TABLESPACE Sale_indx;

Alter table Orders
add constraint FK_Cust_CustID foreign key (custID)
references Customer(CustID);

```

Sample Script

DDL Schema Script for Project

Alter Table with Move on Storage

```
ALTER TABLE hr.admin_emp MOVE
    STORAGE ( INITIAL 20K
              NEXT 40K
              MINEXTENTS 2
              MAXEXTENTS 20
              PCTINCREASE 0 )
    TABLESPACE hr_tbs;

ALTER TABLE hr.admin_emp
    ADD (bonus NUMBER (7,2));
```

Introduction to Database Administration

TOPIC 245

DDL Schema Script including Tables, Columns and Data Types
DDL Schema Script including Tables, Columns and Data Types

DDL Schema Script including Tables, Columns and Data Types

DDL Script for Schema
Sample Create Table - SQL
Sample Create/ Alter Table with Constraints
Sample Create Table with Additional Constraints
DDL Schema Script including Tables, Columns and Data Types

DDL Script for Schema

Scripts are used to create objects (structures) such as tables, sequences, views and PL/SQL programs etc in a schema. Many script contains DDL SQL.

Script can contain DML for populating data

for development and testing.

Similarly, structure of schema objects can be populated using exported DMP file.

DDL Schema Script including Tables, Columns and Data Types

Sample Create Table - SQL

```
CREATE TABLE hr.admin_emp (  
    empno      NUMBER(5) PRIMARY KEY,  
    ename      VARCHAR2(15) NOT NULL,  
    ssn        NUMBER(9) ENCRYPT USING 'AES256',  
    job        VARCHAR2(10),  
    mgr        NUMBER(5),  
    hiredate   DATE DEFAULT (sysdate),  
    photo      BLOB,  
    sal        NUMBER(7,2),  
    hrly_rate  NUMBER(7,2) GENERATED ALWAYS AS (sal/2080),  
    comm       NUMBER(7,2),  
    deptno     NUMBER(3) NOT NULL  
        CONSTRAINT admin_dept_fkey REFERENCES hr.departments  
            (department_id))  
    TABLESPACE admin_tbs  
    STORAGE ( INITIAL 50K);  
  
COMMENT ON TABLE hr.admin_emp IS 'Enhanced employee table';
```

DDL Schema Script including Tables, Columns and Data Types

```

Drop table employee;
Drop table orders;
Drop table customer;
Drop table Item;

create table Customer
(custID          CHAR(10),
CustName        VARCHAR2(25) NOT NULL,
Address         VARCHAR2(50),
City            VARCHAR2(50),
TelNo           VARCHAR2(15),
Email           VARCHAR2(50) UNIQUE)
tablespace TS_Sales;

Alter table Customer
add constraint PK_Custid primary key (CustID);

Create table Orders
(InvoiceNo       CHAR(12),
InvType         CHAR(1),
InvDate         Date NOT NULL,
TotalPrice      Number(6,2),
SalesEmpID      CHAR(5),
CustID          CHAR(8))
tablespace TS_Sales;

Alter table Orders
add constraint PK_InvNo primary key (InvoiceNO)
USING INDEX TABLESPACE Sale_idx;

Alter table Orders
add constraint FK_Cust_CustID foreign key (custID)
references Customer(CustID);

```

Sample Create/ Alter Table with Constraints

DDL Schema Script including Tables, Columns and Data Types

Sample Create Table with Additional Constraints

```

create table Employee
(EmpID          CHAR(3),
EmpName        VARCHAR2(25) NOT NULL,
Address        VARCHAR2(50),
City           VARCHAR2(50),
TelNo          VARCHAR2(15),
Email          VARCHAR2(50) UNIQUE,
Status         CHAR(1))
tablespace TS_Sales;

Alter table Employee
add constraint ck_status
check (Status in ('Y', 'N'));

Alter table Employee
add constraint PK_EmpID primary key (EmpID)
USING INDEX TABLESPACE Sale_idx;

```

TOPIC 246

DDL Schema Script including Allocation of Tablespaces with Sizes

DDL Schema Script including Allocation of Tablespaces with Sizes

DDL Schema Script for Tablespaces

Logical & Physical Storage

Data Tablespaces for Tables

Tablespace for Indexes

Temporary Tablespace for Project

DDL Schema Script including Allocation of Tablespaces with Sizes

Logical & Physical Storage

Creating tablespaces for data, indexes

Tablespaces for table or tables

Tablespace with OS level storage

DDL Schema Script including Allocation of Tablespaces with Sizes

Data Tablespaces for Tables

Create tablespace MedCare Datafile

```
'D:\oracle9i\ora92\oradata\MCare01.dbf' Size 150M online  
MINIMUM EXTENT 64K  
DEFAULT STORAGE (INITIAL 64K NEXT 64K);
```

Create tablespace MCare_IVF Datafile

```
'D:\oracle9i\ora92\oradata\MCIVF01.dbf' Size 50M online  
MINIMUM EXTENT 64K  
DEFAULT STORAGE (INITIAL 64K NEXT 64K);
```

CREATE TABLESPACE MCare_IVFCHART DATAFILE

```
'D:\oracle9i\ora92\oradata\MCare_IVFCHART01.dbf' SIZE 50M  
MINIMUM EXTENT 1024K  
DEFAULT STORAGE (INITIAL 1024K NEXT 1024K);
```

DDL Schema Script including Allocation of Tablespaces with Sizes

Tablespace for Indexes

```
CREATE TABLESPACE MCare_indx  
DATAFILE 'D:\oracle9i\ora92\oradata\MCare_indx01.dbf' SIZE 100M  
MINIMUM EXTENT 64K  
DEFAULT STORAGE (INITIAL 64K NEXT 64K);
```

DDL Schema Script including Allocation of Tablespaces with Sizes

Temporary Tablespace for Project

```
CREATE TEMPORARY TABLESPACE "TEMP_DEFAULT3" TEMPFILE  
'D:\ORACLE9I\ORA92\ORADATA\eahmed\TEMP_DEFAULT3.DBF' SIZE  
1000M EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1M;
```

```
ALTER DATABASE DEFAULT TEMPORARY TABLESPACE "TEMP_DEFAULT3";
```

TOPIC 247

Creation of DML Insert Statements for Data

Creation of DML Insert Statements for Data

DML Schema Script

DDL & DML Script

Sample DML Insert Statements

Sample DML Insert & Update Statements

Creation of DML Insert Statements for Data

DDL & DML Script

One script file can be created to manage DDL & DML commands

Separate files for DDL & DML

DML Insert, Delete and Update commands

All commands are in an order

Well-defined datasets are selected

Creation of DML Insert Statements for Data

```
insert into customer(  
CUSTID,  
CUSTNAME,  
ADDRESS,  
CITY,  
TELNO,  
EMAIL)  
values(  
'88700110',  
'3M Co',  
'41 Main st.',  
'Lahore',  
'0514568877',  
'ejaz@nu.edu.pk'  
);
```

```
insert into customer(  
CUSTID,  
CUSTNAME,  
ADDRESS,
```

Sample DML Insert Statements

Creation of DML Insert Statements for Data

Sample DML Insert & Update Statements

```
Insert into Customer
```

```
...
```

```
Update Sold
```

```
set qty = (select qty  
            from Item  
            where ItemID='254011');
```

TOPIC 248

DDL Schema Script including List of Roles and Privileges

DDL Schema Script including List of Roles and Privileges

Script for Security

Roles & Privileges

Creating User with Grants

DDL Schema Script including List of Roles and Privileges

Roles & Privileges

Create roles

Assign privileges to roles

Assign privileges to users

DDL Schema Script including List of Roles and Privileges

Creating User with Grants

```
create tablespace work
datafile 'E:\oracle\product\10.2.0\db_2\oradata\work01.dbf'size 500M online;
```

```
create user uwork
identified by xyz
default tablespace work
temporary tablespace temp;
```

```
grant connect, resource to uwork;
grant create table to uwork;
```

```
create tablespace work_indx
datafile 'E:\oracle\product\10.2.0\db_2\oradata\WIndx01.dbf'size 200M online;
```

```
create tablespace work_depend
datafile 'E:\oracle\product\10.2.0\db_2\oradata\WDepend01.dbf'size 300M online;
```

DDL Schema Script including List of Roles and Privileges

Creating User with Grants

```
grant select on MED_STOCK to public;
grant select on TRANSACTION to public;
grant select on TRANS_DETAIL to public;
```

```
grant select, insert, update on EMP to public;
grant select, insert, update on DEPT to ejaz;
```

Introduction to Database Administration

TOPIC 249

DDL Schema Script including Synonyms of Tables & Drop of all Objects

DDL Schema Script including Synonyms of Tables & Drop of all Objects

Synonyms & Drop Objects

Generate SQL Script using SELECT Clause

Creating Synonyms

Drop Synonyms

Drop Tables

DDL Schema Script including Synonyms of Tables & Drop of all Objects

Generate SQL Script using Select Clause

```
--Table EMP owner (scott) gives grant:  
grant select on emp to u1;  
--generate the following SQLs from scott schema and run in u1 schema  
select 'create synonym '||tname||' for scott.'||tname||';'  
from tab;
```

DDL Schema Script including Synonyms of Tables & Drop of all Objects

Creating Synonyms

```
grant create synonym to u1;  
grant create public synonym to u1;  
grant create synonym to scott;  
grant create public synonym to scott;
```

DDL Schema Script including Synonyms of Tables & Drop of all Objects

Drop Synonyms

```
--inside same schema of scott:  
create public synonym syn_emp for scott.emp;  
  
drop synonym dept;  
drop synonym emp;
```

DDL Schema Script including Synonyms of Tables & Drop of all Objects

Drop Tables

```
drop table EMP;  
drop table ProjectAssign;  
drop table DEPT;  
drop project;
```

Introduction to Database Administration

TOPIC 250

Revision of Course Contents

Revision of Course Contents

Revision of DBA Course

Introduction to Database & Architecture
Database Design (Development)
DBA
PL/SQL
Project's Preparations & Scripts
Logical & Physical Storage
Revision of Course Contents

Introduction to Database & Architecture

What is Database?
What is DBMS?
ANSI SPARK 3 Level Architecture
2 & 3 Tiers Client Server Architecture
Keys & Constraints
Revision of Course Contents

Database Design (Development)

What is ER Model?
ER Model Artifacts
ER Mapping & Implementation
Revision of Course Contents

DBA

DBA tasks & responsibilities
Data Dictionary
Oracle Installation & Configuration
Logical & Physical Storage
Backup & Recovery
Database Tuning & Performance
Support & Services
Transaction Processing
Revision of Course Contents

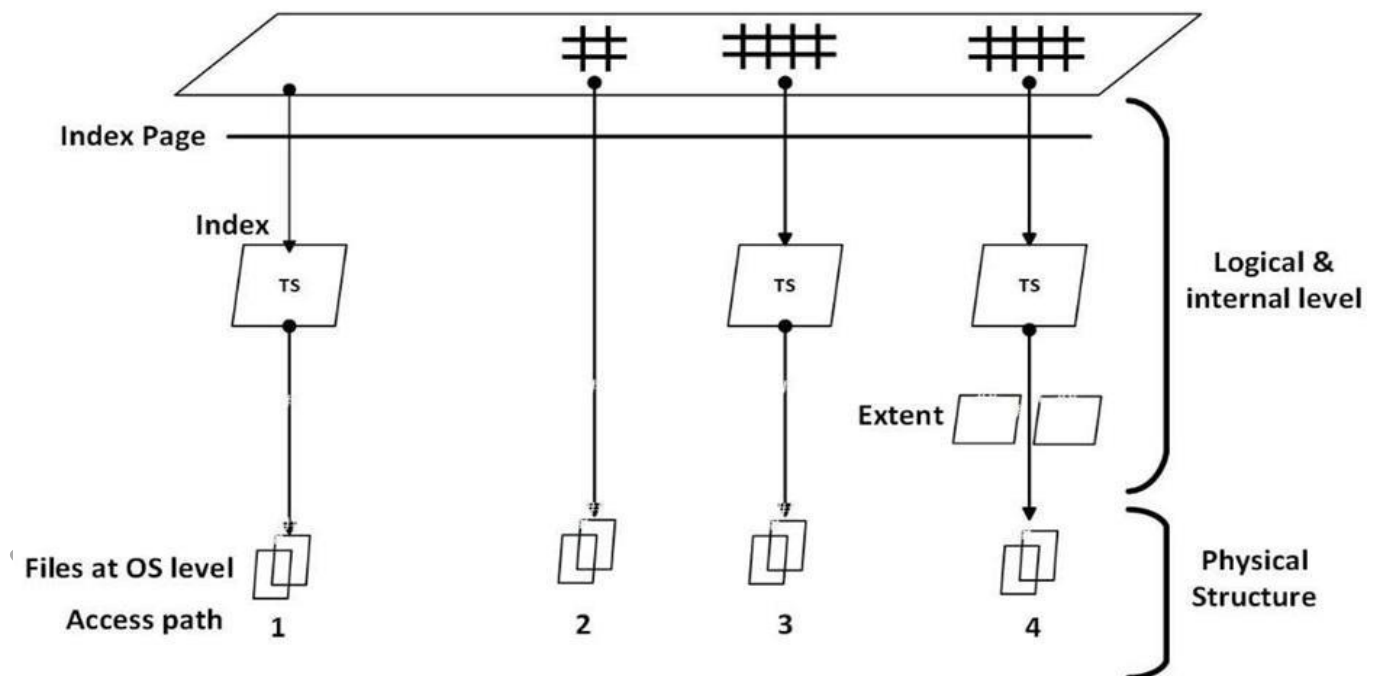
PLSQL

Procedure programming using Blocks
Programming exceptions
Procedures, functions & packages
Cursor and loops
Advance topics using Arrays, Dynamic SQL.
Revision of Course Contents

Project's Preparation and Scripts

Database Mapping or Implementation
Scripts for DDL
Scripts for logical tablespaces
Scripts for Security
Data Migration
Revision of Course Contents

Logical & Physical Storage



THE END