

Week 1:

Data Base user creation, Data definition Language commands, Data Manipulation commands, Data Control Language Commands, Transaction Control Language commands.

Data Base user creation:

```
Create user system identified by manager;  
User created.  
SQL> grant all privilege to system;  
Grant succeeded.
```

Data definition Language commands:

Data Definition Language (DDL) is a subset of SQL and a part of DBMS (Database Management System). DDL consist of Commands to commands like CREATE, ALTER, TRUNCATE and DROP. These commands are used to create or modify the tables in SQL.

CREATE:

This command is used to create a new table in SQL. The user has to give information like table name, column names, and their datatypes.

Syntax :

```
CREATE TABLE table_name  
(  
column_1 datatype,  
column_2 datatype,  
column_3 datatype,  
....  
);
```

Example: –

We need to create a table for storing Student information of a particular College. Create syntax would be as below.

```
CREATE TABLE Student_info  
(  
College_Id number(2),  
College_name varchar(30),  
Branch varchar(10)  
);
```

ALTER:

This command is used to add, delete or change columns in the existing table. The user needs to know the existing table name and can do add, delete or modify tasks easily.

Syntax to add a column to an existing table.

Syntax:

```
ALTER TABLE table_name  
ADD column_name datatype;
```

Example: –

In our Student_info table, we want to add a new column for CGPA. The syntax would be as below as follows.

```
ALTER TABLE Student_info  
ADD CGPA number;
```

TRUNCATE:

This command is used to remove all rows from the table, but the structure of the table still exists.

Syntax:

TRUNCATE TABLE table_name;

Example:

The College Authority wants to remove the details of all students for new batches but wants to keep the table structure. The command they can use is as follows.

TRUNCATE TABLE Student_info;

DROP:

This command is used to remove an existing table along with its structure from the Database.

Syntax: –

Syntax to drop an existing table.

DROP TABLE table_name;

Example –

If the College Authority wants to change their Database by deleting the Student_info Table.

DROP TABLE Student_info;

Data Manipulation commands:

DML is an abbreviation of Data Manipulation Language.

The DML commands in Structured Query Language change the data present in the SQL database.

We can easily access, store, modify, update and delete the existing records from the database using DML commands.

Following are the four main DML commands in SQL:

SELECT Command

INSERT Command

UPDATE Command

DELETE Command

SELECT DML Command:

SELECT is the most important data manipulation command in Structured Query Language. The SELECT command shows the records of the specified table. It also shows the particular record of a particular column by using the WHERE clause.

Syntax of SELECT DML command

Select column_Name_1, column_Name_2,, column_Name_N FROM Name_of_table;

INSERT DML Command:

INSERT is another most important data manipulation command in Structured Query Language, which allows users to insert data in database tables.

Syntax of INSERT Command:

INSERT INTO TABLE_NAME (column_Name1 , column_Name2 , column_Name3 , column_NameN) VALUES (value_1, value_2, value_3, value_N);

UPDATE DML Command:

UPDATE is another most important data manipulation command in Structured Query Language, which allows users to update or modify the existing data in database tables.

Syntax of UPDATE Command:

UPDATE Table_name SET [column_name1= value_1,, column_nameN = value_N] WHERE CONDITION;

DELETE DML Command:

DELETE is a DML command which allows SQL users to remove single or multiple existing records from the database tables.

This command of Data Manipulation Language does not delete the stored data permanently from the database. We use the WHERE clause with the DELETE command to select specific rows from the table.

Syntax of DELETE Command:

DELETE FROM Table_Name WHERE condition;

Data Control Language Commands:

DCL (Data Control Language) includes commands like GRANT and REVOKE, which are useful to give “rights & permissions.” Other permission controls parameters of the database system.

Examples of DCL commands:

Commands that come under DCL:

- Grant
- Revoke

Grant:

This command is use to give user access privileges to a database.

Syntax;

GRANT privilege_name
ON object_name
TO user_name;

Example:

grant select,insert on emp to mybala;

Revoke:

It is useful to back permissions from the user.

Syntax:

REVOKE privilege_name
ON object_name
FROM user_name;

for example:

revoke insert on emp from mybala;

Transaction Control Language commands:

Transaction control language or TCL commands deal with the transaction within the database.

Commit

This command is used to save all the transactions to the database.

Syntax:

Commit;

For example:

```
DELETE FROM Students  
WHERE RollNo =25;  
COMMIT;  
Rollback
```

Rollback command allows you to undo transactions that have not already been saved to the database.

Syntax:

ROLLBACK;

Example:

```
DELETE FROM Students
```

```
WHERE RollNo =25;  
SAVEPOINT
```

This command helps you to sets a savepoint within a transaction.

Syntax:

```
SAVEPOINT SAVEPOINT_NAME;
```

```
SQL> insert into emp  
  2  values(11, 'krishna', 12, 35600)  
  3  ;
```

1 row created.

```
SQL> commit;
```

Commit complete.

```
SQL> update emp  
  2  set sal=999000  
  3  where empno=11  
  4  ;
```

1 row updated.

```
SQL> rollback;
```

Rollback complete.

```
SQL> insert into emp  
  2  values(12, 'janu', 13, 55000);
```

1 row created.

```
SQL> savepoint sp1;
```

Savepoint created.

```
SQL> update emp
  2  set sal=9080
  3  where empno=12;
```

1 row updated.

```
SQL> savepoint sp2;
```

Savepoint created.

```
SQL> rollback to sp1;
```

Rollback complete.

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

EMPNO	EMPNAME	N1
10	bala	12
12	janu	13
11	krishna	12

Week 2:

Database Schema for a customer-sale scenario

Customer (Cust_id : integer, cust_name: string)

Item (item_id: integer, item_name: string, price: integer)

Sale (bill_no: integer, bill_date: date, cust_id: integer, item_id: integer, qty_sold: integer)

For the above schema, perform the following—

- Create the tables with the appropriate integrity constraints
- Insert around 10 records in each of the tables
- List all the bills for the current date with the customer names and item numbers
- List the total Bill details with the quantity sold, price of the item and the final amount
- List the details of the customer who have bought a product which has a price>200
- Give account of how many products have been bought by each customer
- Give a list of products bought by a customer having cust_id as 5
- List the item details which are sold as of today
- Create a view which lists out the bill_no, bill_date, cust_id, item_id, price, qty_sold, amount
- Create a view which lists the daily sales date wise for the last one week

Aim: Create the tables with the appropriate integrity constraints and Insert around 10 records in each of the tables

a) SQL>create table customer(cid number(4) primary key, cname varchar2(15) not null);

Name	Null?	Type
-----	-----	-----
CID	NOT NULL	NUMBER(4)
CNAME	NOT NULL	VARCHAR2(15)

SQL> create table item (itemid number(4) primary key, item_name varchar2(20) unique, price number(5));

Name	Null?	Type
-----	-----	-----

ITEMID	NOT NULL	NUMBER(4)
ITEM_NAME		VARCHAR2(20)
PRICE		NUMBER(5)

SQL>create table sale(billno number(5) primary key, billdate date, cid number(4) references customer(cid), itemid number(4) references item(itemid), quantity_sold varchar2(10));

SQL> desc sale

Name	Null?	Type
-----	-----	-----
BILLNO	NOT NULL	NUMBER(5)
BILLDATE		DATE
CID		NUMBER(4)
ITEMID		NUMBER(4)
QUANTITY_SOLD		VARCHAR2(10)

(b) insert into customer values(10,'kumar');

SQL> select * from customer;

CID	CNAME
-----	-----
1	hari
2	pavan
3	damani
4	datri
5	damayanthi
6	suma
7	kamala
8	deepu
9	vijay
10	kumar

SQL> delete from item where itemid=1111;

1 row deleted.

SQL> select * from item;

ITEMID	ITEM_NAME	PRICE
1000	burger	22
1001	rice	42
1002	salt	13
1003	sugar	43
1004	jagerry	67
1005	mirchi	7
1006	tamarind	37
1007	pepper	33
1008	hideandseek	35
1009	mariegold	24
1010	bangle	3

b) insert into sale values(2009,'30-oct-2015',3,1002,'4kg');

SQL> select * from sale;

BILLNO	BILLDATE	CID	ITEMID	QUANTITY_S
2000	12-JAN-15	1	1000	2kg
2001	12-JAN-13	2	1001	1kg
2002	01-FEB-14	3	1002	half kg
2003	23-MAR-12	4	1010	4kg
2004	13-APR-10	4	1009	5kg
2005	13-APR-10	5	1008	3kg

2006	10-MAY-11	7	1003	44kg
2007	13-DEC-11	8	1005	44kg
2008	01-NOV-12	9	1006	4kg
2009	30-OCT-15	3	1002	4kg

c) List all the bills for the current date with the customer names and item numbers

SQL>insert into sale values(2011,'16-jul-2015',1,1009,'3kg');

SQL>select c.cid,c.cname,s.billno from customer c,item i,sale s where c.cid=s.cid and i.itemid=s.itemid and s.billdate=to_char(sysdate);

CID	CNAME	BILLNO
-----	-----	-----
1	hari	2011

d) List the total Bill details with the quantity sold, price of the item and the final amount

SQL>select s.billno,s.billdate,s.quantity_sold,i.price, (i.price*s.quantity_sold) from item i,sale s

where i.itemid=s.itemid;

e)List the details of the customer who have bought a product which has a price<20

SQL>select c.cid,c.cname,i.item_name,i.price from customer c,item i,sale s where c.cid=s.cid and i.itemid=s.itemid and i.price<20;

CID	CNAME	ITEM_NAME	PRICE
-----	-----	-----	-----
3	damani	salt	13
4	datri	bangle	3
8	deepu	mirchi	7
3	damani	salt	13

f) Give a count of how many products have been bought by each customer

SQL>select c.cid,count(i.itemid)from customer c,item i,sale s where c.cid=s.cid and i.itemid=s.itemid group by(c.cid);

CID	COUNT(I.ITEMID)
-----	-----------------

-----	-----
1	2
2	1
4	2
5	1
8	1
3	2
7	1
9	1

SQL>select c.cid,i.item_name,c.cname,count(i.itemid)from customer c,item i,sale s where c.cid=s.cid and i.itemid=s.itemid group by(c.cid,c.cname,i.item_name);

CID	ITEM_NAME	CNAME	COUNT(I.ITEMID)
-----	-----	-----	-----
5	hideandseek	damayanthi	1
7	sugar	kamala	1
2	rice	pavan	1
8	mirchi	deepu	1
1	mariegold	hari	1
3	salt	damani	2
1	burger	hari	1
4	bangle	datri	1
4	mariegold	datri	1
9	tamarind	vijay	1

g) Give a list of products bought by a customer having cust_id as 5

SQL>select i.itemid,i.item_name,i.price,c.cid,c.cname from customer c,item i,sale s where c.cid=s.cid and i.itemid=s.itemid and s.cid=5

ITEMID	ITEM_NAME	PRICE	CID	CNAME
1008	hideandseek	35	5	damayanthi

h) List the item details which are sold as of today

SQL>select i.itemid,i.item_name,i.price from item i,sale s where s.itemid=i.itemid and s.billdate='16-jul-2015';

ITEMID	ITEM_NAME	PRICE
1009	mariegold	24

i) Create a view which lists out the bill_no, bill_date, cust_id, item_id, price, qty_sold, amount

SQL>create view cis_view as(select s.billno,s.billdate,c.cid,i.itemid,i.price, s.quantity_sold, (i.price*s.quantity_sold) amount from customer c,item i,sale s where c.cid=s.cid and i.itemid=s.itemid)

View created.

j) Create a view which lists the daily sales date wise for the last one week

SQL>create view cis_view as(select s.billno,s.billdate,c.cid,i.itemid,i.price, s.quantity_sold, (i.price*s.quantity_sold) amount from customer c,item i,sale s where c.cid=s.cid and i.itemid=s.itemid) and s.billdate>'12-jan-2015';

View created.

Week 3:

Database Schema for a Student Library scenario

Student(Stud_no : integer, Stud_name: string)

Membership(Mem_no: integer, Stud_no: integer)

Book(book_no: integer, book_name:string, author: string)

**issue(iss_no:integer, iss_date: date, Mem_no: integer,
book_no: integer)**

For the above schema, perform the following—

- Create the tables with the appropriate integrity constraints
- Insert around 10 records in each of the tables
- List all the student names with their membership numbers
- List all the issues for the current date with student and Book names
- List the details of students who borrowed book whose author is CJDATE
- Give a count of how many books have been bought by each student
- Give a list of books taken by student with stud_no as 5
- List the book details which are issued as of today
- Create a view which lists out the iss_no, iss_date, stud_name, book name
- Create a view which lists the daily issues-date wise for the last one week

AIM: Create the tables with the appropriate integrity constraints

Insert around 10 records in each of the tables

a) create table student(sid number(3) primary key, sname varchar2(15));

SQL> desc student;

Name	Null?	Type
SID	NOT NULL	NUMBER(3)
SNAME		VARCHAR2(15)

SQL> insert into student values (10,'rahul');

1 row created.

SQL> select * from student;

SID	SNAME
-----	-------

-----	-----
1	deepu
2	lavanya
3	ujjwala
4	kiran
5	kamala
6	vijay
7	ratnam
8	mani
9	ravi
10	rahul

SQL> create table membership(mid number(4) primary key, sid number(3) references student(sid));

Table created.

SQL> desc membership;

Name	Null?	Type
-----	-----	-----
MID	NOT NULL	NUMBER(4)
SID		NUMBER(3)

SQL> insert into membership values(100,1);

10 rows selected.

SQL> select * from membership;

MID	SID
-----	-----
100	1
101	2
102	3

103	4
104	5
105	6
106	7
107	8
108	9
109	10

SQL> create table book(bid number(4) primary key, bname varchar2(15), bauthor varchar2(30));

Table created.

SQL> desc book;

Name	Null?	Type
-----	-----	-----
BID	NOT NULL	NUMBER(4)
BNAME		VARCHAR2(15)
BAUTHOR		VARCHAR2(30)

SQL> select * from book;

BID	BNAME	BAUTHOR
-----	-----	-----
2000	se	pressman
2001	stm	boris
2002	dbms	korth
2003	ppl	andrew
2004	cn	tanenbaum
2005	os	galvin
2006	uml	booch
2007	c	vara

2008	www	lee
2009	co	archi
2010	android	Richard

SQL> create table issue(issueno varchar2(10) primary key, issuedate date, mid number(3) references membership(mid), bid number(4) references book(bid));

Table created.

SQL> desc issue;

Name	Null?	Type
-----	-----	-----
ISSUENO	NOT NULL	VARCHAR2(10)
ISSUEDATE		DATE
MID		NUMBER(3)
BID		NUMBER(4)

b) SQL> insert into issue values('a10','12-apr-2015',102,2006);

SQL> select * from issue;

ISSUENO	ISSUEDATE	MID	BID
-----	-----	-----	-----
3000	22-JUL-15	100	2000
3001	02-JAN-13	101	2001
3002	12-FEB-15	102	2003
3003	02-MAR-15	103	2004
3004	29-JUL-15	104	2005
3005	09-JUL-15	106	2006
3006	19-JUL-15	107	2009
3007	09-AUG-14	108	2010
3008	19-APR-14	109	2008
3009	09-MAY-15	108	2007

3010 09-MAY-15 105 2003

c) List all the student names with their membership numbers

SQL> select s.sname,m.mid,s.sid from student s,membership m where s.sid=m.sid;

SNAME	MID	SID
-----	-----	-----
deepu	100	1
lavanya	101	2
ujjwala	102	3
kiran	103	4
kamala	104	5
vijay	105	6
ratnam	106	7
mani	107	8
ravi	108	9
rahul	109	10

d) List all the issues for the current date with student and Book names

**SQL> select s.sname,b.bname,i.issuedate from student s,membership m,book b,issue i
where s.sid=m.sid and m.mid=i.mid and b.bid=i.bid and i.issuedate=to_char(sysdate);**

SNAME	BNAME	ISSUEDATE
-----	-----	-----
kamala	os	29-JUL-15

e) List the details of students who borrowed book whose author is Neekhilesh

**SQL> select s.sid,s.sname,b.bauthor,i.issuedate,i.issueno
from student s,book b,membership m,issue i
where s.sid=m.sid and m.mid=i.mid and b.bid=i.bid and b.bauthor='pressman';**

SID	SNAME	BAUTHOR	ISSUEDATE	ISSUENO
-----	-----	-----	-----	-----
1	deepu	pressman	22-JUL-15	3000

f) Give a count of how many books have been bought by each student

**SQL> select s.sid,count(distinct s.sid) from student s,membership m,book b,issue i
where s.sid=m.sid and m.mid=i.mid and b.bid=i.bid group by(s.sid);**

SID	COUNT(DISTINCTS.SID)
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

g) Give a list of books taken by student with stud_no as 5

**SQL> select b.bid,b.bname,i.issuedate from student s,membership m,book b,issue i
where s.sid=m.sid and m.mid=i.mid and b.bid=i.bid and s.sid=5;**

BID	BNAME	ISSUEDATE
2005	os	29-JUL-15
2000	se	29-JUL-15

h) List the book details which are issued as of today

**SQL> select b.bid,b.bname,i.issuedate from book b,issue i
where b.bid=i.bid and i.issuedate=to_char(sysdate);**

BID	BNAME	ISSUEDATE
2005	os	29-JUL-15
2000	se	29-JUL-15

i) Create a view which lists out the iss_no, iss_date, stud_name, book name

**SQL> create view smbi_view as(select s.sname,s.sid,i.issueno,b.bname from student
s,book b,issue i,membership m where s.sid=m.sid and m.mid=i.mid and b.bid=i.bid);**

View created.

SQL> select * from smbi_view;

SNAME	SID	ISSUENO	BNAME
deepu	1	3000	se
lavanya	2	3001	stm
ujjwala	3	3002	ppl
kiran	4	3003	cn
kamala	5	3004	os

ratnam	7	3005	uml
mani	8	3006	co
ravi	9	3007	android
rahul	10	3008	www
ravi	9	3009	c
vijay	6	3010	ppl

SNAME	SID	ISSUENO	BNAME
-----	-----	-----	-----
kamala	5	3012	se

12 rows selected.

j) Create a view which lists the daily issues-date wise for the last one week

**SQL> create view did as(select i.issueno,i.issuedate from issue i
where issuedate between '20-jul-2015' and '31-jul-2015');**
View created.

SQL> select * from did;

ISSUENO	ISSUEDATE
-----	-----
3000	22-JUL-15
3004	29-JUL-15
3012	29-JUL-15

Week 4:

Database Schema for a Employee-pay scenario

employee(emp_id : integer, emp_name: string)

department(dept_id: integer, dept_name:string)

paydetails(emp_id : integer, dept_id: integer, basic: integer,

deductions: integer, additions: integer, DOJ: date)

payroll(emp_id : integer, pay_date: date)

For the above schema, perform the following—

- Create the tables with the appropriate integrity constraints
- Insert around 10 records in each of the tables
- List the employee details department wise
- List all the employee names who joined after particular date
- List the details of employees whose basic salary is between 10,000 and 20,000
- Give a count of how many employees are working in each department
- Give a names of the employees whose netsalary>10,000
- List the details for an employee_id=5
- Create a view which lists out the emp_name, department, basic, dedeuctions, netsalary
- Create a view which lists the emp_name and his netsalary

a) SQL> create table employee(eid number(2) primary key, ename varchar2(15) not null unique);

SQL> desc employee;

Name	Null?	Type

EID	NOT NULL	NUMBER(2)
ENAME	NOT NULL	VARCHAR2(15)

SQL> insert into employee values(10,'kiran');

SQL> select * from employee;

EID	ENAME

1	deepu
2	naresh
3	ujj
4	lavanya
5	pavan
6	paritosh
7	harsha
8	swaga
9	venkat
10	kiran

```
SQL> create table department(depid number(4) primary key,
2 dept_name varchar2(10));
```

```
SQL> desc department;
```

Name	Null?	Type

DEPID	NOT NULL	NUMBER(4)
DEPT_NAME		VARCHAR2(10)

```
SQL> select * from department;
```

DEPID	DEPT_NAME

100	cse
101	it
102	ece
103	ce
104	mec
105	chem
106	eee

107 ins
 108 hs
 109 biology
 110 humanities

SQL> create table paydetails(

2 eid number(2) references employee(eid) primary key,

3 depid number(3) references department(depid),

4 deductions number(4) not null,

5 additions number(5) not null

Doj date);

L> select * from paydetails;

EID	DEPID	BASIC	DEDUCTIONS	ADDITIONS	DOJ
2	101	7000	300	405	12-JUN-14
3	102	7000	300	405	12-JUN-14
1	101	6000	202	400	12-JAN-14
4	103	6000	305	400	12-FEB-14
5	104	6005	355	300	12-MAR-15
6	105	9005	111	100	31-JUL-15
7	107	8005	177	154	29-JUL-15
8	107	5305	127	124	01-APR-15
9	109	3305	117	114	11-APR-15
10	110	3505	157	164	11-AUG-14

SQL>

SQL> desc payroll;

Name	Null?	Type
-----	-----	-----

EID NOT NULL NUMBER(2)

PAYDATE DATE

SQL> select * from payroll;

EID PAYDATE

1	30-JAN-15
2	30-JAN-15
3	27-FEB-15
4	27-MAR-15
5	07-JUN-15
6	17-JAN-15
7	07-JUL-15
8	08-MAR-15
9	08-MAR-15
10	8-MAR-15

c) List the employee details department wise

**SQL> select e.eid,e.ename,d.depid,d.dept_name
2 from employee e,department d,paydetails p
3 where e.eid=p.eid and d.depid=p.depid;**

EID	ENAME	DEPID	DEPT_NAME
1	deepu	101	it
2	naresh	101	it
3	ujj	102	ece
4	lavanya	103	ce
5	pavan	104	mec
6	paritosh	105	chem
8	swaga	107	ins
7	harsha	107	ins
9	venkat	109	biology
10	iran	110	humanities

d) List all the employee names who joined after particular date

**SQL> select e.eid,e.ename,p.doj,d.dept_name
2 from employee e,department d,paydetails p**

3 where e.eid=p.eid and d.depid=p.depid and p.doj>'20-jan-2014';

EID	ENAME	DOJ	DEPT_NAME
2	naresh	12-JUN-14	it
3	ujj	12-JUN-14	ece
4	lavanya	12-FEB-14	ce
5	pavan	12-MAR-15	mec
6	paritosh	31-JUL-15	chem
8	swaga	01-APR-15	ins
7	harsha	29-JUL-15	ins
9	venkat	11-APR-15	biology
10	iran	11-AUG-14	humanities

e) List the details of employees whose basic salary is between 10,000 and 20,000

SQL> select e.eid,e.ename,d.dept_name,d.depid,p.basic

2 from employee e,department d,paydetails p

3 where e.eid=p.eid and d.depid=p.depid and p.basic between 2000 and 7000;

EID	ENAME	DEPT_NAME	DEPID	BASIC
2	naresh	it	101	7000
1	deepu	it	101	6000
3	ujj	ece	102	7000
4	lavanya	ce	103	6000
5	pavan	mec	104	6005
8	swaga	ins	107	5305
9	venkat	biology	109	3305
10	kiran	humanities	110	3505

f) Give a count of how many employees are working in each department

SQL> select count(eid),depid from paydetails p group by(depid);

COUNT(EID)	DEPID
1	102
1	110
2	101
2	107
1	104
1	105

1 109

1 103

g) Give a names of the employees whose netsalary>10,000

**SQL> select e.ename from employee e where e.eid in(
2 select e.eid from paydetails p where p.eid=e.eid and (p.basic-p.deductions)
>7000);**

ENAME

paritosh

harsha

h) List the details for an employee_id=5

SQL> select * from employee where eid=5;

EID ENAME

5 pavan

i) Create a view which lists out the emp_name, department, basic, dedeuctions, netsalary

**SQL> create view edp_view as(select e.ename,e.eid,d.depuid,p.basic,p.deductions,(
p.basic-p.deductions) netsalary from
2 employee e,department d,paydetails p
3 where e.eid=p.eid and d.depuid=p.depuid);**

View created.

SQL> select * from edp_view;

ENAME EID DEPID BASIC DEDUCTIONS NETSALARY

naresh 2 101 7000 300 6700

ujj 3 102 7000 300 6700

deepu 1 101 6000 202 5798

lavanya 4 103 6000 305 5695

pavan 5 104 6005 355 5650

paritosh 6 105 9005 111 8894

harsha 7 107 8005 177 7828

swaga 8 107 5305 127 5178

venkat 9 109 3305 117 3188

kiran	10	110	3505	157	3348
-------	----	-----	------	-----	------

SQL> select * from enet_view;

EID	ENAME	NETSALARY
2	naresh	6700
3	ujj	6700
1	deepu	5798
4	lavanya	5695
5	pavan	5650
6	paritosh	8894
7	harsha	7828
8	swaga	5178
9	venkat	3188
10	kiran	3348

Week 5:

Database Schema for a student-Lab scenario

Class(class: string, descrip: string)

Student(stud_no: integer, stud_name: string, class: string)

Lab(mach_no: integer, Lab_no: integer, description: String)

Allotment(Stud_no: Integer, mach_no: integer, dayof week: string)

For the above schema, perform the following—

- Create the tables with the appropriate integrity constraints.
- Insert around 10 records in each of the tables.
- List all the machine allotments with the student names, lab and machine numbers.
- List the total number of lab allotments day wise.
- Give a count of how many machines have been allocated to the 'CSIT' class.
- Give a machine allotment details of the stud_no 5 with his personal and class details.
- Count for how many machines have been allocated in **Lab_no 1** for the day of the week as "Monday".
- How many students class wise have allocated machines in the labs.
- Create a view which lists out the stud_no, stud_name, mach_no, lab_no, dayofweek.
- Create a view which lists the machine allotment details for "Thursday".

SQL> create table class(cname varchar2(10) primary key, cdesc varchar2(20));

SQL> create table studen(sid number(3) primary key, sname varchar2(10),cname varchar2(10) references class(cname));

SQL> create table lab(macid number(5) primary key,labid varchar2(10),ldesc varchar2(10));

SQL> create table allotment(sid number(3) references studen(sid),macid number(5) references lab(macid),day varchar2(10));

SQL> desc class;

Name	Null?	Type

CNAME	NOT NULL	VARCHAR2(10)
CDESC		VARCHAR2(20)

SQL> desc studen;

Name	Null?	Type

SID	NOT NULL	NUMBER(3)
SNAME		VARCHAR2(10)
CNAME		VARCHAR2(10)

SQL> desc lab;

Name	Null?	Type

MACID	NOT NULL	NUMBER(5)
LABID		VARCHAR2(10)
LDESC		VARCHAR2(10)

SQL> desc allotment;

Name	Null?	Type

SID		NUMBER(3)
MACID		NUMBER(5)
DAY		VARCHAR2(10)

SQL> select * from class;

CNAME	CDESC

1a	cse
1b	it
1c	ece
1d	cse
2d	chem
2c	civil
2b	eee
2a	cse
3a	it
3b	civil
3c	pharma

CNAME	CDESC

3d	bpharma
4a	eie
4b	cse
4c	cse
4d	ese

16 rows selected.

SQL> select * from studen;

SID	SNAME	CNAME
100	hari	1a
101	mani	1b
102	pari	1c
103	pandu	1a
104	pinky	1c
105	paritosh	1d
106	harish	2d
107	cutie	2c
108	deepu	2b
109	ujj	2a
110	lav	4a

SID	SNAME	CNAME
111	madhu	2a
112	kari	2c

SQL> select * from lab;

MACID	LABID	LDESC
1	lab-1	ibm
2	lab-2	oracle
3	lab-3	mic
4	lab-4	pg
5	lab-1	ibm
6	lab-2	oracle
7	lab-3	mic
8	lab-4	pg
9	cc1	first
10	cc2	first

SQL> select * from allotment;

SID	MACID	DAY
100	1	sat
101	2	mon

103	4	tue
103	4	tue
105	5	wed
106	6	thu
107	7	fri
108	8	sat
108	8	sat
109	5	sat
110	2	tue

c) List all the machine allotments with the student names, lab and machine numbers

```
SQL> select l.macid,s.sname,l.labid,s.sid,c.cname
2 from class c,student s,lab l,allotment a
3 where c.cname=s.cname and s.sid=a.sid and l.macid=a.macid;
```

MACID	SNAME	LABID	SID	CNAME
1	hari	lab-1	100	1a
2	lav	lab-2	110	4a
2	mani	lab-2	101	1b
4	pandu	lab-4	103	1a
4	pandu	lab-4	103	1a
5	ujj	lab-1	109	2a
5	paritosh	lab-1	105	1d
6	harish	lab-2	106	2d
7	cutie	lab-3	107	2c
8	deepu	lab-4	108	2b
8	deepu	lab-4	108	2b

11 rows selected.

d) List the total number of lab allotments day wise

```
SQL> select a.macid,l.labid,l.ldesc,a.day from lab l,allotment a where
2 l.macid=a.macid;
```

MACID	LABID	LDESC	DAY
1	lab-1	ibm	sat
2	lab-2	oracle	mon
4	lab-4	pg	tue
4	lab-4	pg	tue
5	lab-1	ibm	wed
6	lab-2	oracle	thu
7	lab-3	mic	fri
8	lab-4	pg	sat
8	lab-4	pg	sat
5	lab-1	ibm	sat
2	lab-2	oracle	tue

e) Give a count of how many machines have been allocated to the 'CSIT' class

SQL> select count(macid) from allotment where sid in(select sid from studen where cname='4a');

COUNT(MACID)

1

f) Give a machine allotment details of the stud_no 5 with his personal and class details

SQL> select a.macid,a.sid,s.sname,c.cname,c.cdsc

2 from class c,studen s,lab l,allotment a

3 where c.cname=s.cname and s.sid=a.sid and l.macid=a.macid and s.sid='101'

MACID	SID	SNAME	CNAME	CDESC
2	101	mani	1b	it

g) Count for how many machines have been allocated in Lab_no 1 for the day of the week as "Monday"

SQL> select count(a.macid) from allotment a,lab l where l.macid=a.macid and a.day='sat' and l.labid='lab-1';

COUNT(A.MACID)

2

h) How many students class wise have allocated machines in the labs

SQL> select count(sid) allocated_students_in_lab

2 from studen s where sid in(select sid from allotment)group by(cname);

ALLOCATED_STUDENTS_IN_LAB

1
2
1
1
1
1
1

i) Create a view which lists out the stud_no, stud_name, mach_no, lab_no, dayofweek

SQL> create view sl as(select s.sid,s.sname,l.macid,l.labid,a.day

2 from studen s,lab l,allotment a

3 where s.sid=a.sid and l.macid=a.macid);

View created.

SQL> select * from sl;

SID	SNAME	MACID	LABID	DAY
100	hari	1	lab-1	sat
101	mani	2	lab-2	mon
103	pandu	4	lab-4	tue
103	pandu	4	lab-4	tue
105	paritosh	5	lab-1	wed
106	harish	6	lab-2	thu
107	cutie	7	lab-3	fri
108	deepu	8	lab-4	sat
108	deepu	8	lab-4	sat
109	ujj	5	lab-1	sat
110	lav	2	lab-2	tue

11 rows selected.

j) Create a view which lists the machine allotment details for “Thursday”.

**SQL> create view st as(select a.macid,l.labid,a.day,s.sid from studen s,lab l,al
lotment a where s.sid=a.sid and l.macid=a.macid and a.day='thu');**

View created.

SQL> select * from st;

MACID	LABID	DAY	SID
6	lab-2	thu	106

Week 6:

Database Schema for a Video Library scenario

Customer(cust_no: integer,cust_name: string)

Membership(Mem_no: integer, cust_no: integer)

Cassette(cass_no:integer, cass_name:string, Language: String)

Iss_rec(iss_no: integer, iss_date: date, mem_no: integer, cass_no: integer)

For the above schema, perform the following—

- a) Create the tables with the appropriate integrity constraints**
- b) Insert around 10 records in each of the tables**
- c) List all the customer names with their membership numbers**
- d) List all the issues for the current date with the customer names and cassette names**
- e) List the details of the customer who has borrowed the cassette whose title is — The Legend**
- f) Give a count of how many cassettes have been borrowed by each customer**
- g) Give a list of book which has been taken by the student with mem_no as 5**
- h) List the cassettes issues for today i) Create a view which lists out the iss_no, iss_date, cust_name, cass_name**
- j) Create a view which lists issues-date wise for the last one week**

AIM: Create the tables with the appropriate integrity constraints Insert around 10 records in each of the tables

SQL>create table customer

(cust_no number(5) primary key,

cust_name varchar2(20));

SQL>desc customer;

Name Null? Type

.....

CUST_NO NOT NULL NUMBER(5)

CUST_NAME VARCHAR2(20)

SQL>insert into customer values(&cust_no,"&cust_name");

SQL>select * from customer;

CUST_NO CUST_NAME

.....

50 scott

51 pandey

52 varshney

53 naidu

54 bhimbira

SQL>create table membership(mem_no number(5) primary key,cust_no number(5)
references customer(cust_no));

SQL>dsc membership;

Name Null? Type

.....

MEM_NO NOT NULL NUMBER(5)

CUST_NO NUMBER(5)

SQL>insert into memshipvalues(&mem_no,&cust_no);

SQL>select * from memship;

MEM_NO CUST_NO

.....

920 50

981 51

897 52

820 53

928 54

SQL>create table cassette(cass_no number(5) primary key, Cass_namevarchar2(15),language
varchar2(15));

```
SQL>desc cassette;
```

```
Name Null? Type
```

```
.....
```

```
CASS_NO NOT NULL NUMBER(5)
```

```
CASS_NAME VARCHAR2(15)
```

```
LANGUAGE VARCHAR2(15)
```

```
SQL>insert into cassette values(&cass_no,&cass_name,&language);
```

```
SQL>select * from cassette;
```

```
CASS_NO CASS_NAME LANGUAGE
```

```
.....
```

```
1 tagore telugu
```

```
2 the lion king English
```

```
3 anniyan tamil
```

```
4 indra telugu
```

```
5 lord of rings English
```

```
SQL>create table issu_rec(iss_no number(5) primary key,iss_datedate,mem_no  
number(5)references memship(mem_no),cass_no number(5) references cassette(cass_no));
```

```
SQL>desc issu_rec;
```

```
Name Null? Type
```

```
.....
```

```
ISS_NO NOT NULL NUMBER(5)
```

```
ISS_DATE DATE
```

```
MEM_NO NUMBER(5)
```

```
CASS_NO NUMBER(5)
```

```
SQL>select * from issu_rec;
```

```
ISS_NO ISS_DATE MEM_NO CASS_NO
```

```
.....
```

```
22 07-JAN-06 920 1
```

23 10-JAN-00 981 2

26 10-JAN-06 897 5 3

01-JAN-06 820 4

34 31-DEC-05 928 3

c) List all the customer names with their membership numbers

```
SQL>select c.custname,m.memno from customer1 c, membership1 m where
c.custno=m.custno;
```

CUSTNAME MEMNO

.....

NIKHIL 51

VIVEK 52

SHRAVAN 58

VAMSI 57

SHIVA 56

d)List all the issues for the current date with the customer names and cassette names

```
SQL>select i.issno,c.custname,cc.cassettename
```

```
from customer1 c,membership1 m,cassette cc,issrec1 I
```

```
where i.issdate=to_char(sysdate) and c.custno=m.custno and i.cassno=cc.cassno and
i.memno=m.memno;
```

OutPut: no rows selected.

e) List the details of the customer who has borrowed the cassette whose title is “ The Legend”

```
SQL> select c.cid,c.cname
```

```
from customer c, videpmem m, videoissue I,cassette t
```

```
where c.cid=m.cid and m.memid=i.memid and i.cassid=t.cassid and t.cassname=“The
Legend”
```

cid cname

.....

1 The Legend

f) Give a count of how many cassettes have been borrowed by each customer

```
SQL>select v.memid,count(v.memid)
```

```
from customer c, videomem m,cassette c, videoissue v
```

```
where c.cid=m.cid and m.memid=v.memid and c.cassid=v.cassid
```

```
group by(v.memid);
```

```
MEMID COUNT(V.MEMID)
```

```
100 1
```

```
108 1
```

```
102 2
```

```
110 1
```

```
101 1
```

```
107 1
```

```
104 2
```

```
109 1
```

```
106 1 9
```

```
rows selected.
```

g) Give a list of book which has been taken by the student with mem_no as 5

```
SQL> select c.cassid,c.cassname,c.language,v.issueno,v.issuedate
```

```
from customer c, videomem m,cassette c, videoissue v
```

```
where c.cid=m.cid and m.memid=v.memid and c.cassid=v.cassis and m.memid=102;
```

```
CASSID CASSNAME LANGUAGE ISSUENO ISSUEDATE
```

```
-----
```

```
102 os telugu 2002 30-JUL-15
```

```
102 os telugu 2003 04-AUG-15
```

h) List the cassettes issues for today

```
SQL> select c.cassid,c.cassname,c.language,v.issueno,v.issuedate
```

```
from customer c, videomem m,cassette c, videoissue v
```

where c.cid=m.cid and m.memid=v.memid and c.cassid=v.cassis and
v.issuedate=to_char(sysdate);

CASSID CASSNAME LANGUAGE ISSUENO ISSUEDATE

1006 java telugu 2006 09-DEC-15

i) Create a view which lists out the iss_no, iss_date, cust_name, cass_name

SQL> create view cmav as(select c.cassid,c.cassname,c.language,v.issueno,v.issuedate
from customer c, videomem m,cassette c, videoissue v
where c.cid=m.cid and m.memid=v.memid and a.cassid=v.cassis);

SQL> select * from cmav;

CNAME CASSNAME ISSUENO ISSUEDATE

hari se 2002 30-JUL-15

pavan spm 2003 04-AUG-15

damani os 2004 30-AUG-15

damani java 2002 30-JUL-15

datri c 2001 30-AUG-15

suma ds 2002 22-JUL-15

kamala oops 2002 23-AUG-15

7 records selected.

j) Create a view which lists issues-date wise for the last one week

SQL> create view cmav as(select v.issueno,v.issuedate, a.cassid, c.cassname
from customer c, videomem m,cassette a, videoissue v

where c.cid=m.cid and m.memid=v.memid and a.cassid=v.cassis and v.issuedate between
"10 jul-2014" and "18-aug-2015");

ISSUENO ISSUEDATE CASSID CASSNAME

2002 30-JUL-15 1000 se

2003 04-AUG-15 1001 spm

2004 30-AUG-15 1002 se

2002 30-JUL-15 1003 os

2001 30-AUG-15 1004 os

2002 22-JUL-15 1005 os

2002 23-AUG-15 1006 os