

## SQL Queries

1. Consider the following relations:

S (S#, SNAME, STATUS, CITY)

SP (S#, P, QTY)

P (P#, PNAME, COLOR, WEIGHT, CITY)

Give an expression in SQL for each of queries below:

- (i) Get supplier names for supplier who supply at least one red part
- (ii) Get supplier names for supplier who do not supply part P2.

**Ans:(i)** SELECT SNAME FROM S  
WHERE S# IN (SELECT S# FROM SP  
WHERE P# IN (SELECT P# FROM P  
WHERE COLOR = RED'))

**(ii)** SELECT SNAME FROM S  
WHERE S# NOT IN (SELECT S# FROM SP WHERE P# = 'P2')

2. Construct a view for the above relations which has the information about suppliers and the parts they supply. The view contains the S#, SNAME, P# , PNAME renamed as SNO, NAME, PNO, PNAME.

**Ans:** CREATE VIEW SUP\_PART (SNO, NAME, PNO, PNAME) AS  
SELECT S.S#, SNAME, P.P#, PNAME  
FROM S, SP, P  
WHERE S.S# = SP.S# AND P.P# = SP.P#

3. Consider the following relational schemas:

EMPLOYEE (EMPLOYEE\_NAME, STREET, CITY)

WORKS (EMPLOYEE\_NAME, COMPANYNAME, SALARY)

COMPANY (COMPANY\_NAME, CITY)

- (i) Find the names of all employees who work for 'First Bank Corporation'.
- (ii) Find the names and company names of all employees sorted in ascending order of company name and descending order of employee names of that company.
- (iii) Change the city of First Bank Corporation to 'New Delhi'

**Ans: (i)** SELECT EMPLOYEE\_NAME  
FROM WORKS  
WHERE COMPANYNAME = 'First Bank Corporation';

**(ii)** SELECT EMPLOYEE\_NAME, COMPANYNAME  
FROM WORKS  
ORDER BY COMPANYNAME, EMPLOYEE\_NAME DESC;

**(iii)** UPDATE COMPANY  
SET CITY = 'New Delhi'  
WHERE COMPANY\_NAME = 'First Bank Corporation';

4. Consider the following relational database:

STUDENT (name, student#, class, major)

COURSE (course name, course#, credit hours, department)

SECTION (section identifier, course#, semester, year, instructor)

GRADE\_REPORT (student#, section identifier, grade)

PREREQUISITE (course#, prerequisite#)

Specify the following queries in SQL on the above database schema.

- (i) Retrieve the names of all students majoring in 'CS' (Computer Science).
- (ii) Retrieve the names of all courses taught by Professor King in 1998
- (iii) Delete the record for the student whose name is 'Smith' and whose student number is 17.
- (iv) Insert a new course <'Knowledge Engineering', 'CS4390', 3, 'CS'>

**Ans:** (i) SELECT NAME FROM STUDENT WHERE MAJOR = 'CS'

(ii) SELECT COURSE\_NAME FROM COURSE C, SECTION S  
WHERE C.COURSE# = S.COURSE#  
AND INSTRUCTOR = 'KING' AND YEAR = 1998  
OR  
SELECT COURSE\_NAME FROM COURSE  
WHERE COURSE# IN (SELECT COURSE# FROM SECTION  
WHERE INSTRUCTOR = 'KING' AND YEAR = 1998)

(iii) DELETE FROM STUDENT WHERE NAME = 'Smith' AND STUDENT# = 17

(iv) INSERT INTO COURSE  
VALUES ('Knowledge Engineering', 'CS4390', 3, 'CS')

5. Given the following relations

TRAIN (NAME, START, DEST)

TICKET (PNRNO., START, DEST, FARE)

PASSENGER (NAME, ADDRESS, PNRNO.)

Write SQL expressions for the following queries:

**Note:** Assume NAME of Train is a column of Ticket.

- (i) List the names of passengers who are travelling from the start to the destination station of the train.
- (ii) List the names of passengers who have a return journey ticket.
- (iii) Insert a new Shatabdi train from Delhi to Bangalore.
- (iv) Cancel the ticket of Tintin.

**Ans:** (i) SELECT P.NAME FROM TRAIN T, TICKET I, PASSENGER P  
WHERE P.PNRNO = I.PNRNO AND T.NAME = I.NAME  
AND T.START = I.START AND T.DEST = I.DEST

(ii) SELECT NAME FROM PASSENGER  
WHERE PNRNO IN (SELECT DISTINCT A.PNRNO  
FROM TICKET A, TICKET B WHERE A.PNRNO = B.PNRNO  
AND A.START = B.DEST AND A.DEST = B.START)

(iii) INSERT INTO TRAIN  
VALUES('Shatabdi', 'Delhi', 'Bangalore')

(iv) DELETE FROM TICKET  
WHERE PNRNO = (SELECT PNRNO FROM PASSENGER  
WHERE NAME = 'Tintin')

6. Consider the relations defined below:

PHYSICIAN (regno, name, telno, city)

PATIENT (pname, street, city)

VISIT (pname, regno, date\_of\_visit, fee)

Where the regno and pname identify the physician and the patient uniquely

respectively. Express queries (i) to (iii) in SQL.

- (i) Get the name and regno of physicians who are in Delhi.
- (ii) Find the name and city of patient(s) who visited a physician on 31 August 2004.
- (iii) Get the name of the physician and the total number of patients who have visited her.

**Ans:** (i) Select name, regno from PHYSICIAN where city = 'Delhi';  
(ii) Select pname, city from PATIENT,VISIT where PATIENT.pname=VISIT.pname  
anddate\_of\_visit = '31-Aug-04';  
(iii) select name, count(\*) from PHYSICIAN, VISIT  
wherePHYSICIAN.regno = VISIT.regno group by  
Physician .regno;

7. Consider the following relations:

BRANCH(bno, street, area, city, pcode, Tel\_no, Fax\_no)  
STAFF(Sno, Fname, Lname, address, position , salary, bno)  
Express the following queries in SQL:

- (i) List the staff who work in the branch at '163 main street'
- (ii) Find staff whose salary is larger than the salary of every member of staff at branch B3.

**Ans:** (i) Select Fname, Lname from STAFF, BRANCH where STAFF.bno =  
BRANCH.bno and street = '163 main street'  
(ii) Select Fname, Lname from STAFF where salary >( select max (salary) from  
Staff where bno='B3')

8. (i) Consider employee (e\_no, e\_name, e\_salary, d\_code), dept (d\_code, d\_name) and dependent (depndt\_name, e\_no, relation). Show the names of employees in purchase and accounts departments with at least one dependent.  
(ii) Consider student (std\_id, std\_name, date\_of\_birth, phone, dept\_name). Put the data for a student with student id200, name arun, birth date 1st February, 1985, phonenumber (01110 32818 and dept name English in the student table.  
(iii) A constraint named less\_than\_20 was defined on the field date\_of\_birth of tablestudent. Delete this constraint.  
(iv) Consider the table student and list names of students in the departments other than  
(v) Consider employee table of (i) and list names of department(s) for which averagemaths and computer.salary for department is more than 10,000.  
(vi) Create role named role\_table that allows a user to create tables. Using role\_tableallow users kripa and reena to create tables.  
(vii) Create a view emp\_dep containing e\_name and number of dependents from the tables employee and dependent of (i)

**Ans:**(i) SELECT e\_name FROM employee, dependent, dept  
WHERE employee.d\_code=dept.d\_code  
AND employee.e\_no=dependant.e\_no  
AND d\_name IN ('ACCOUNTS', 'PURCHASE')  
GROUP BY e\_name  
HAVING COUNT (e\_no)>=1  
(ii) INSERT INTO student VALUES (200,'arun','01-FEB-85','(0111)32818','English');  
(iii) ALTER TABLE student DROP CONSTRAINT less\_than\_20;  
(iv) SELECT std\_name FRM student WHERE dept\_name NOT IN('Maths','Computer);  
(v) (SELECT d\_name FROM dept  
WHERE d\_code IN (SELECT d\_code  
FROM emp GROUP BY d\_code  
HAVING AVG(sal) ? 10000);

```
(vi) CREATE ROLE role_table;
      GRANT CREATE ANY TABLE TO role_table;
      GRANT role_table TO kripa, reena;
(vii) CREATE VIEW emp_dept AS SELECT ename,
      COUNT(*) FROM employee, dependent
      WHERE employee.e_no = dependant.e_no GROUP BY e_name;
```

9. Consider the relations given below

Borrower (id\_no, name)

Book (accno., title, author, borrower\_idno)

(a) Define the above relations as tables in SQL making real world assumptions about the type of the fields. Define the primary keys and the foreign keys.

(b) For the above relations answer the following queries in SQL

What are the titles of the books borrowed by the borrower whose id-no is 365.

(i) Find the numbers and names of borrowers who have borrowed books on DBMS in ascending order in id\_no.

(ii) List the names of borrowers who have borrowed at least two books.

**Ans: a)** Create table Book

```
(Accno int Primary Key,
title char(30),
author char(30),
borrow-idno int references Borrower_idno );
Create table borrower
(id-no int Primary Key,
name char(30) );
```

**b) (i)** Select title from Book, Borrower where Borrower.id\_no = Book.borrower-idno and borrower.id\_no = 365

**(ii)** Select id\_no, name from Borrower, Book where Borrower.id\_no = Book.borrower\_idno and title = 'DBMS' order by id\_no asc;

**(iii)** Select name from Borrower, Book where Borrower.id\_no = book.borrower\_id\_no having count (\*) > 2;

10. Consider the following relations with keys underlined

Street (name, location, city)

House (number, street\_name)

Lives (name, house\_number)

Define the above relations as tables in SQL making real world assumptions about the type of the fields. Define the primary keys and the foreign keys.

**Ans:** Create table street (name character(30) primary key, location character(30), city character(30));

Create table house( number integer primary key, street\_name character(30) references street(name));

Create table lives ( name character(30) primary key, house number integer references house(number));

11. For the relations given in Q11 answer the following queries in SQL

(i) Get the names of persons who live in the street named 'Mahatma Gandhi'.

(ii) Get the house numbers street wise.

(iii) Get the numbers of houses which are not occupied

**Ans: (i)** Select lives.name from Street, House, Lives where Street.name =

House.street\_name and House. Number = Lives. House\_number and street\_name = 'Mahatma Gandhi';

(ii) Select number from House  
order by street\_name;

(iii) Select number from house  
minus  
Select house\_number from lives;

**12.** Consider the relations

EMP(ENO,ENAME,AGE,BASIC\_SALARY)

WORK\_IN(ENO,DNO)

DEPT(DNO,DNAME,CITY)

Express the following queries in SQL

(i) Find names of employees who work in a deptt. in Delhi.

(ii) Get the deptt. number in which more than one employee is working.

(iii) Find name of employee who earns highest salary in 'HR' department.

**Ans:**(i) select ENAME from EMP, WORK\_IN, DEPT where EMP.ENO=  
WORK\_IN.ENO and WORK\_IN.DNO= DEPT.DNO and CITY= 'Delhi';

(ii) select DNO from WORK\_IN group by DNO having count(\*) >1

(iii) select ENAME from EMP e where BASIC\_SALARY >= (select  
max(BASIC\_SALARY) from DEPT,WORK\_IN where DNAME =  
'HR' and e.ENO = WORK\_IN.ENO and WORK\_IN. DNO=  
DEPT.DNO)

### Joins

**Example1:** Suppose we want to fetch records from two tables emp, dept where deptno of employee is equal to dept no of dept.

SELECT \* FROM EMP E,DEPT D WHERE E.DEPTNO=D.DEPTNO;

**Example2:** Suppose there are two tables category and product.Both table contains category\_id. We want to fetch the records where both tables contain same category\_id .

SELECT \* FROM CATEGORIES C ,PRODUCTS P where P.CATEGORYID = C.CATEGORYID;

**Example3:** Suppose there are two table Employee and Bonus. We want to fetch records where both tables contain same empno.

SELECT \* FROM EMPLOYEE E,BONUS B WHERE E.EMPNO=B.EMPNO;

**Example4:** Suppose there are two tables Employee and Salary\_Grade.Salary\_Grade contains lowsals,Hisal.We want to fetch all records where sal of employees less than lowsals.

SELECT \* FROM EMPLOYEE E, SALARY\_GRADE SD WHERE E.SAL<=SD.LOWSAL.

**Example5:** There are two table emp and dept. We want to fetch records of emp where deptno between 10 and 20.

SELECT \* FROM EMP E, DEPT D WHERE D.DEPTNO BETWEEN 10 AND 20;

**Example6:** There are two table Emp and Emp\_details. Emp\_details contains emp\_no,f\_name,L\_name,City,Pin,Dob. Fetch the empno,f\_name,sal,city where sal>=4000.

```
SELECT  
E.EMPNO,ED.SAL,ED.F_NAME,ED.CITY,ED.PIN,ED.DOB  
FROM EMP E,EMP_DETAILS ED WHERE E.SAL>=4000;
```

**Example7:** From the last examle fetch the records from Emp table whose city in between DELHI,BANGALORE,PUNE.

```
SELECT * FROM EMP WHERE EMP_DETAILS.CITY  
IN("DELHI","BANGALORE","PUNE");
```

**Example8:** In Emp table there are two columns empno and mgr. Fetch the records where empno and mgr are same.

```
SELECT * FROM EMP E1, EMP E2 WHERE E1.EMPNO=E2.MGR
```

**Example9:** In Emp\_SAL table there are two columns gross\_pay and net\_pay. Fetch the records where gross\_pay and net\_pay are same.

```
SELECT * FROM EMP_SAL E1,EMP_SAL E2 WHERE  
E1.GROSS_PAY=E2.NET_PAY
```

**Example10:** InEmployee\_details table retrieve the record whose city are same.

```
SELECT * FROM EMPLOYEE_DETAILS E1,EMPLOYEE_DETAILS E2 WHERE  
E1.CITY=E2.CITY
```