**Thapar Institute of Engineering & Technology, Patiala**
Computer Science and Engineering Department
**MID-SEMESTER EXAMINATION (Solution)**

| | |
|---|---|
| Course Code: UCS310 | Course Name: Database Management System |
| B.E. (Second Year), Semester-II | Branch: COE, CSE |
| March 11, 2024 | Monday, 3 PM – 5 PM |
| Time: 2 Hours, M. Marks: 30 | Name of Faculty: RKR, MK, RRN, SKH, CHP, SRO, AAD, VAP |

**Q1.** (1M for explanation and 0.5M for example)

**(a)** External/view/user level:
It is the users' view of the database. This level describes that part of the database that is relevant to each user. For example, one user may view dates in the form (day, month, year), while another may view dates as (year, month, day).

Conceptual/ Logical level: It is the community view of the database. This level describes what data is stored in the database and the relationships among the data.

Physical/internal level: It is the physical representation of the database on the computer. This level describes how the data is stored in the database.

**(b)** Multiple row subquery returns one or more rows to the outer SQL statement. You may use the IN, ANY, or ALL operator in outer query to handle a subquery that returns multiple rows.

Example: SELECT employee_id, e_name, job_id, salary
FROM employees
WHERE salary IN (SELECT salary FROM employees WHERE job_id = 'IT_PROG')

You can write subqueries that return multiple columns. The following example retrieves the order amount with the lowest price, group by agent code.

Example: Select product_id, qty from item where (product_id, qty) IN (Select product_id, qty from item where order_id = 600);

**(c)** Entity integrity: is concerned with ensuring that each row of a table has a unique and non-null primary key value; this is the same as saying that each row in a table represents a single instance of the entity type modelled by the table.
Example: Primary Key

Referential integrity**:** is a term used in database design to describe the relationship between two tables. It is important because it ensures that all data in a database remains consistent and up to date. It helps to prevent incorrect records from being added, deleted, or modified.
Example: Foreign key

**(d)** At least any three differences

(i) Structure: The file system is a way of arranging the files in a storage medium within a computer. DBMS is software for managing the database.

(ii)Data Redundancy: Redundant data can be present in a file system. In DBMS there is no redundant data.

(iii)Backup and Recovery: It doesn't provide Inbuilt mechanism for backup and recovery of data if it is lost. It provides in house tools for backup and recovery of data even if it is lost.

(iv)Consistency: There is less data consistency in the file system. There is more data consistency because of the process of normalization.

Examples: File system: Excel sheet DBMS: MongoDB

**Q2. a)** $(AB)^+ = \{ABCDE\}$ **0.5M**

$(BC)^+ = \{ABCDE\}$ **0.5M**

**b)** 'B' is the candidate key. **1M (steps + correct answer), 0.5M(only for correct answer)**

Non-prime attributes are {A, C, D, E} **1M (No partial mark)**

**c)** B→A, B→C, AB→C, BC→A **(0.5M for each)**

**Q3**. 2 Marks for each correct query. No partial marking

Suppliers (*sid:* integer, *sname:* string, *address:* string)
Parts (*pid:* integer, *pname:* string, *color:* string)
Catalog(*sid:* integer, *pid:* integer, *cost:* real)

**a)** Find the *sid*s of suppliers who supply some red parts or lives at "221 Packer Street (using subquery).

**SQL:**

```
SELECT S.sid
FROM Suppliers S
WHERE S.address = '221 Packer street'
OR S.sid IN ( SELECT C.sid
              FROM Parts P, Catalog C
              WHERE P.color='red' AND P.pid = C.pid )
```

**b)** Find names of suppliers who supply at least one part (using co-related subquery).

**SQL:**

```
SELECT S.sname
FROM Suppliers S
WHERE EXISTS (SELECT C.sid
              FROM Catalog C where C.sid= S.sid);
```

**c)** Find the list of suppliers who supply red coloured parts.

**SQL:**

```
SELECT S.SID, S.sname
FROM Suppliers S, Catalog C, Parts P
WHERE S.sid = C.sid and C.pid = P.pid
                and P.color = 'Red';
```

**d)** Display the details of suppliers whose all supply cost is more than 10000 and supplying less than five parts.

**SQL:**

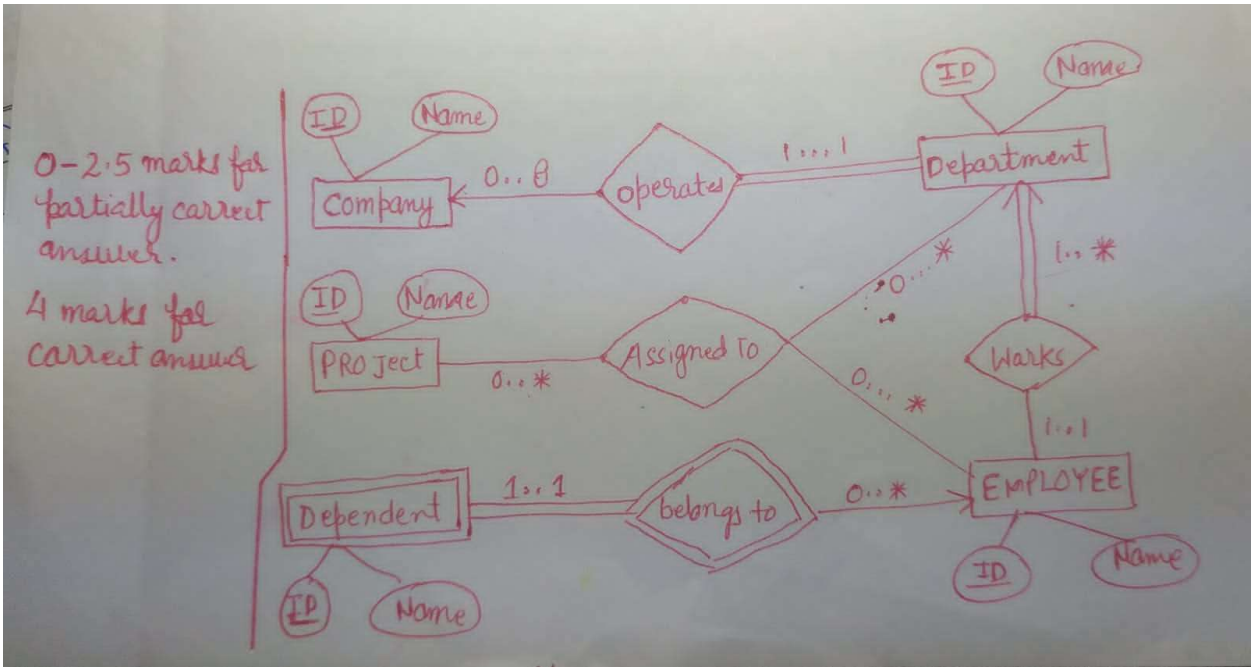
```
SELECT * from Suppliers
where sid IN (select sid from catalog
                        group by sid
                        having sum(cost)>= 10000 and count(*)<5);
```

**e)** Display the names of suppliers except 'Harish' and 'Dinesh' who supply the lowest priced parts.

**SQL:**

```
Select S.Sname, C.Cost from Suppliers S, Catalog C
where S.sid= C.sid and C.Cost = (Select min(cost) from Catalog C1 where C1.sid= S.sid and
S.sname NOT IN ('Harish', 'Dinesh'));
```

**Q4. a)**



**Q4.b)**

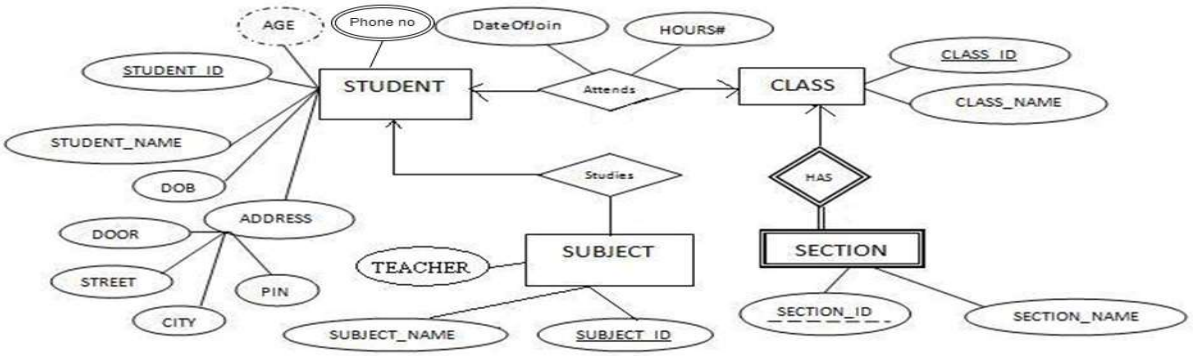



**Table STUDENT (Student_ID, Student_Name, DOB, Door, Street, City, Pin)** **1M**

**Table STUDENT_PHONE (Phone no, Student_ID)** **1M**

**Student_ID is the foreign key refers STUDENT table**

**Table CLASS (Class_ID, Class_Name, Student_ID, DateOfJoin, Hours#)** **1M**

***Student_ID is the foreign key refers STUDENT table***

**Table SUBJECT (Subject_ID, Subject_Name, Teacher, Student_ID)** **1M**

***Student_ID is the foreign key refers STUDENT table***

**Table SECTION (Section_ID, Class_ID, Section_Name)** **1M**

***Class_ID is the foreign key refers CLASS table***

(OR)

**Table CLASS (Class_ID, Class_Name)** **1M**

**Table STUDENT (Student_ID, Student_Name, DOB, Door, Street, City, Pin, Class_ID, DateOfJoin, Hours#)** **1M**

***Class_ID is the foreign key refers CLASS table***

**Table STUDENT_PHONE (Phone no, Student_ID)** **1M**

**Student_ID is the foreign key refers STUDENT table**

**Table SUBJECT (Subject_ID, Subject_Name, Teacher, Student_ID)** **1M**

***Student_ID is the foreign key refers STUDENT table***

**Table SECTION (Section_ID, Class_ID, Section_Name)** **1M**

***Class_ID is the foreign key refers CLASS table***