IARE '24 DISCORD TEAM

II-II

DBMS MODULE 3 SOLUTIONS

KAVYA SREE • UJJWAL

SQL QUERY - BASICS , RDBMS - NORMALIZATION

Many Pending

DBMS MODULE 3

PART A

1 Write the SQL expression for the following Queries. Sailor Schema (sailor id, Sailorname, Rating.Age) Reserves (Sailor id, Boat id, Day) Boat Schema (Boat id, Boatname.color) 1. Find the names of sailors who have reserved boat name 103;

```
SELECT S.sname
 FROM Sailors S, Reserves R
WHERE S.sid = R.sid AND R.bid = 103
```

2. Find the sailor id of sailors who have reserved a red boat;

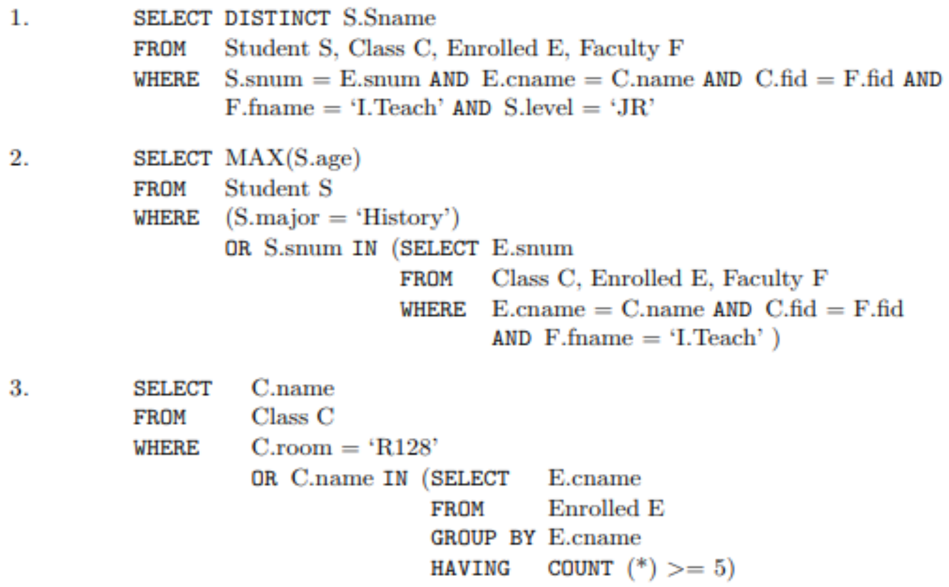
```
SELECT S.sid,
FROM Sailors S, Reserves R, Boats B
WHERE S.sid = R.sid AND R.bid = B.bid AND B.color = ‘red’
```

3. Find the colors of boats reserved by the sailor rubber?

4. Find the names of sailors who have reserved a red boat?

```
SELECT S.sname,
FROM Sailors S, Reserves R, Boats B
WHERE S.sid = R.sid AND R.bid = B.bid AND B.color = ‘red’
```

2 For the following relational database, give the expressions in SQL. student(stuno, stuname, major,level,age) Class(Classname, meets at, Room, fid) Faculty(fid,fname,deptid) 1. Find the names of all uniors (level = JR) Who are enrolled in a class taught by I.Teach? 2. Find the age of the oldest student who is either a history major or is enrolled in a course taught by I.Tech? 3. Find the names of all classes that either meet in room R128 or have five or more students enrolled?



```
1.      SELECT DISTINCT S.Sname
        FROM   Student S, Class C, Enrolled E, Faculty F
        WHERE  S.snum = E.snum AND E.cname = C.name AND C.fid = F.fid AND
               F.fname = 'I.Teach' AND S.level = 'JR'

2.      SELECT MAX(S.age)
        FROM   Student S
        WHERE  (S.major = 'History')
               OR S.snum IN (SELECT E.snum
                             FROM   Class C, Enrolled E, Faculty F
                             WHERE  E.cname = C.name AND C.fid = F.fid
                                    AND F.fname = 'I.Teach' )

3.      SELECT   C.name
        FROM     Class C
        WHERE    C.room = 'R128'
                 OR C.name IN (SELECT    E.cname
                               FROM      Enrolled E
                               GROUP BY  E.cname
                               HAVING    COUNT (*) >= 5)
```

3 Write the SQL expressions for the following relational database. sailor schema (sailor id, Boat id, sailorname, rating, age) Recerves (Sailor id, Boat id, Day) Boat Schema (boat id, Boatname, color)

1. Find the age of the youngest sailor for each rating level?

SELECT S.sname, S.age

FROM Sailors S

WHERE S.age <= ALL (SELECT age FROM Sailors)

2. Find the age of the youngest sailor who is eligible to vote for each rating level with at lead two such sailors? 3. Find the No.of reservations for each red boat?

For all sailors queries refer Tutorial_6.doc (toronto.edu)

4 Consider the following schema: Suppliers(sid: integer, sname: string, address: string) Parts(pid: integer, pname: string, color: string) Catalog(sid: integer, pid: integer, cost: real) The Catalog relation lists the prices charged for parts by Suppliers. Answer the following questions: Give an example of an updatable view involving one relation. 1. Give an example of an updatable view involving two relations. 2. Give an example of an insertable-into view that is updatable. 3. Give an example of an insertable-into view that is not updatable.

5 Consider following relations in DB and solve the queries: Student (GR NO, name, gender, address, city, class) Marks (GR NO, sub1, sub2, sub3, total, per) 1. Display the student of 'FYBCA' and 'TYBCA'. (2 mark each) 2. Display the marks of student whose gr no> 100. 3. Count the no of girls in FYBCA. 4. count the no: of first class students in TYBCA

6. Consider a relation scheme R = (A, B, C, D, E, H) on which the following functional dependencies hold: A→ B, BC→ D, E→ C, D→ A. Write the candidate keys of R.

R = (A, B, C, D, E, H)

$$\left.\begin{array}{l} A \longrightarrow B \\ BC \longrightarrow D \\ E \longrightarrow C \\ D \longrightarrow A \end{array}\right\} \begin{array}{l} \text{Consider RHS} \\ \text{Missing elements are} \\ \text{possible Candidate keys} \end{array}$$

E, H are missing. Try combinations that make R.

$[EA]^+ \longrightarrow$ E, A, B, C, D (Not possible)

$[EAH]^+ \longrightarrow$ E, A, H, B, C, D (Candidate key)

$[EBH]^+ \longrightarrow$ E, B, H, C, D, A (Candidate key)

$(EDH)^+ \longrightarrow$ E, D, H, C, A, B (Candidate key

★ AEH, BEH, DEH are Candidate keys

7.Consider the following relational schemes for a library database: Book (Title, Author, Catalog no, Publisher, Year, Price) Collection (Title, Author, Catalog no) the following are functional dependencies: Title Author → Catalog no Catalog no → Title Author Publisher Year Publisher Title Year → Price Assume Author, Title is the key for both schemes. Apply the appropriate normal form for Book Cancellation.

Book - bcnf (so it is 3nf ,2nf,1nf as well)

Collection - 3nf

Explanation :

- The table “Collection” is in BCNF as there is only one functional dependency “Title Author ⟶ Catalog_no” and {Author, Title} is key for collection.
- Book is not in BCNF because Catalog_no is not a key and there is a functional dependency “Catalog_no ⟶ Title Author Publisher Year”.
- Book is not in 3NF because non-prime attributes (Publisher Year) are transitively dependent on key [Title, Author].
- Book is in 2NF because every non-prime attribute of the table is either dependent on the whole of a candidate key [Title, Author], or on another non-prime attribute.

In the table book, candidate keys are {Title, Author} and {Catalog_no}. In table Book, non-prime attributes (attributes that do not occur in any candidate key) are Publisher, Year and Place

8. Consider a schema R (A, B, C, D) and functional dependencies A → B and C →D. Solve and find whether the decomposition of R into R1 (A, B) and R2(C, D) belongs to which one or both (dependency preserving and lossless join)?

Dependency preserving but not lossless join

Explanation: Dependency Preserving Decomposition**:**

Decomposition of R into R1 and R2 is a dependency preserving decomposition if closure of functional dependencies after decomposition is same as closure of FDs before decomposition. A simple way is to just check whether we can derive all the original FDs from the FDs present after decomposition.

In the above question R(A, B, C, D) is decomposed into R1 (A, B) and R2(C, D) and there are only two FDs A → B and C → D. So, the decomposition is dependency preserving.

Lossless-Join Decomposition:

Decomposition of R into R1 and R2 is a lossless-join decomposition if at least one of the following functional dependencies are in F+ (Closure of functional dependencies)

In the above question R(A, B, C, D) is decomposed into R1 (A, B) and R2(C, D), and R1 ∩ R2 is empty. So, the decomposition is not lossless.

9. Consider the relation R(A,B,C,D,E,F) and FDs A →BC, F→ A, C→ AD → E, E → D AD is the decomposition of R into R1(A,C,D) R2 (B,C,D) and R3 (E,F,D) loss less? Explain the requirement of Lossless decomposition.

FDs are not proper in the question.

The decomposition is lossless when it satisfies the following statement –

- If we union the sub Relation R1 and R2 then it must contain all the attributes that are available in the original relation R before decomposition.
- Intersections of R1 and R2 cannot be Null. The sub relation must contain a common attribute. The common attribute must contain unique data.

10. Suppose the schema R(A,B,C,D,E) is decomposed into (A,B,C) and (A,D,E)show that the decomposition is not a dependency preserving decomposition if the following set of FD hold A→ BC, CD→ E, B→ D, E →A

R1 ∩ R2 = A; (A → BC) (A → ABC) (R1 ∩ R2 → R1) this is a lossless-join decomposition.

PART B

1 **Define a View in SQL. Write about updates on views.**

Views in SQL

- Views in SQL are considered as a virtual table. A view also contains rows and columns.
- To create the view, we can select the fields from one or more tables present in the database.
- A view can either have specific rows based on certain condition or all the rows of a table.

1. Creating view

A view can be created using the CREATE VIEW statement. We can create a view from a single table or multiple tables.

Syntax:

```
CREATE VIEW view_name AS
SELECT column1, column2.....
FROM table_name
WHERE condition;
```

3. Creating View from multiple tables

View from multiple tables can be created by simply include multiple tables in the SELECT statement.

4. Deleting View

A view can be deleted using the Drop View statement.

Syntax

```
DROP VIEW view_name;
```

For more information refer DBMS SQL View - javatpoint

2 Illustrate Group by and Having clauses with examples.

1. Having Clause :

Having Clause is basically like the aggregate function with the GROUP BY clause. The HAVING clause is used instead of WHERE with aggregate functions. While the GROUP BY Clause groups rows that have the same values into summary rows. The having clause is used with the where clause in order to find rows with certain conditions. The having clause is always used after the group By clause.

```
SELECT COUNT (SALARIES) AS COUNT_SALARIES, EMPLOYEES
FROM EMPLOYEES
GROUP BY SALARIES
HAVING COUNT(SALARIES) > 1;
```

2. Group By Clause :

The GROUP BY clause is often used with aggregate functions (MAX, SUM, AVG) to group the results by one or more columns or In simple words we can say that The GROUP BY clause is used in collaboration with the SELECT statement to arrange required data into groups.

The GROUP BY statement groups rows that have the same values. This Statement is used after the where clause. This statement is often used with some aggregate function like SUM, AVG, COUNT atc. to group the results by one or more columns.

```
SELECT COUNT (SALARIES) AS COUNT_SALARIES, EMPLOYEES
FROM EMPLOYEES
GROUP BY SALARIES;
```

3 Discuss about Complex integrity constraints in SQL

4 Write a nested query to find the names of sailors who have reserved both a red and green boat. Write a nested query to find the names of sailors who have reserved all boats.

5 Discuss various DML statements in SQL and explain with Examples.

It is used for accessing and manipulating data in a database. It handles user requests.

Select: It is used to retrieve data from a database.

Insert: It is used to insert data into a table.

Update: It is used to update existing data within a table.

Delete: It is used to delete all records from a table.

Merge: It performs UPSERT operation, i.e., insert or update operations.

Call: It is used to call a structured query language or a Java subprogram

6 define referential integrity constraint, unique key. Is unique +not null is same as primary key

7 What are nested queries? What is correlation in nested queries? Explain

8 Consider the following schema instructor (ID, name, dept name), teaches (ID, course id, sec id, semester, year), section (course id, sec id, semester, year), student (ID, name, dept name), takes (ID, course id, sec id, semester, year, grade) Write the following queries in SQL 1. Find the names of the students not registered in any section 2. Find the total number of courses taught department wise 3. Find the total number of courses registered department wise

9

10 Define functional dependencies. How are primary keys related to FDs?

The functional dependency is a relationship that exists between two attributes. It typically exists between the primary key and non-key attribute within a table.

X → Y

The left side of FD is known as a determinant, the right side of the production is known as a dependent.

For example:

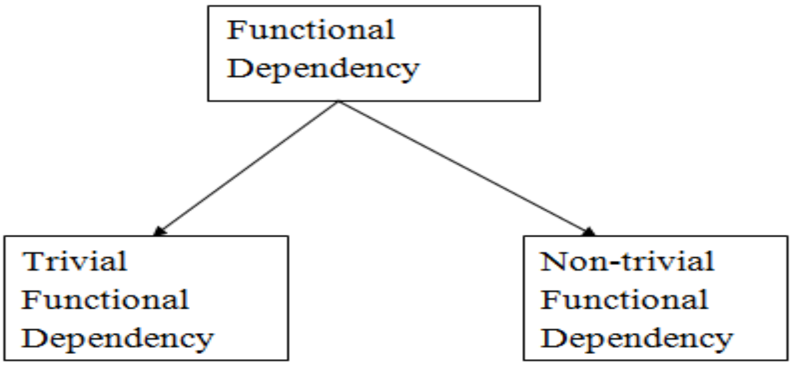
Assume we have an employee table with attributes: Emp_Id, Emp_Name, Emp_Address.

Here Emp_Id attribute can uniquely identify the Emp_Name attribute of employee table because if we know the Emp_Id, we can tell that employee name associated with it.

Functional dependency can be written as:

1. Emp_Id → Emp_Name

We can say that Emp_Name is functionally dependent on Emp_Id.


Functional Dependency
Trivial Functional Dependency
Non-trivial Functional Dependency

1. Trivial functional dependency

- A → B has trivial functional dependency if B is a subset of A.
- The following dependencies are also trivial like: A → A, B → B

2. Non-trivial functional dependency

- A → B has a non-trivial functional dependency if B is not a subset of A.
- When A intersection B is NULL, then A → B is called as complete non-trivial.

Refer DBMS Functional Dependency - javatpoint

11. Define normalization? Explain 1NF, 2NF, 3NF Normal forms

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies.
- Normalization divides the larger table into smaller and links them using relationships.
- The normal form is used to reduce redundancy from the database table.
- The main reason for normalizing the relations is removing these anomalies

1NF	A relation is in 1NF if it contains an atomic value.
2NF	A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key.
3NF	A relation will be in 3NF if it is in 2NF and no transition dependency exists.

Extra info visit : DBMS Normalization: 1NF, 2NF, 3NF and BCNF with Examples - javatpoint

12. Describe properties of decompositions ?

Properties of Relational Decomposition

1. **Attribute Preservation**
   The attributes in R will appear in at least one relation schema Ri in the decomposition, i.e., no attribute is lost. This is called the Attribute Preservation condition of decomposition.

2. **Dependency Preservation**
   If each functional dependency X→Y specified in F appears directly in one of the relation schemas Ri in the decomposition D or could be inferred from the dependencies that appear in some Ri. This is the Dependency Preservation.

3. **Non Additive Join Property**
   Another property of decomposition that D should possess is the Non Additive Join Property, which ensures that no spurious tuples are generated when a NATURAL JOIN operation is applied to the relations resulting from the decomposition.

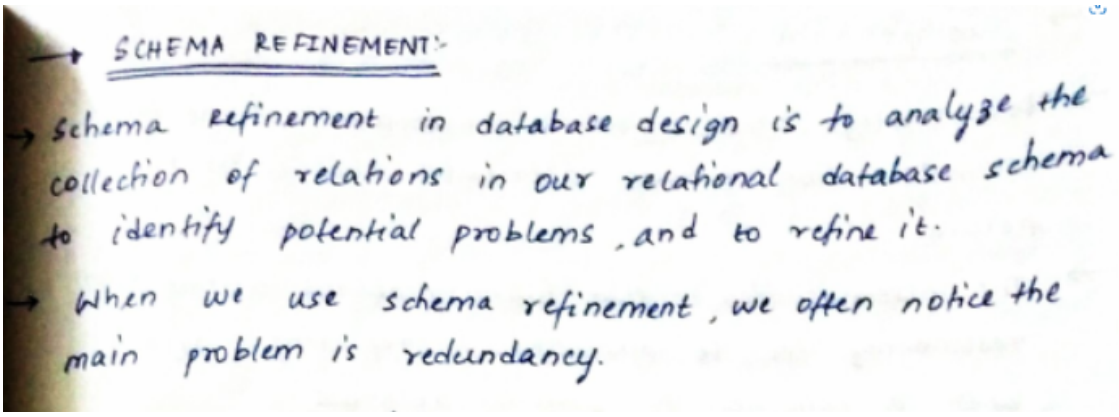
4. **No redundancy**
   Decomposition is used to eliminate some of the problems of bad design like anomalies, inconsistencies, and redundancy.If the relation has no proper decomposition, then it may lead to problems like loss of information.

5. **Lossless Join**

Lossless join property is a feature of decomposition supported by normalisation. It is the ability to ensure that any instance of the original relation can be identified from corresponding instances in the smaller relations.

13. Explain about Schema refinement in Database design ?

Normalisation or Schema Refinement is a technique of organising the data in the database. It is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like Insertion, Update and Deletion Anomalies.



→ SCHEMA REFINEMENT:-

→ Schema refinement in database design is to analyze the collection of relations in our relational database schema to identify potential problems, and to refine it.

→ When we use schema refinement, we often notice the main problem is redundancy.

Problems Caused by Redundancy

Storing the same information redundantly, that is, in more than one place within a database, can lead to several problems:

Redundant storage: Some information is stored repeatedly.

Update anomalies: If one copy of such repeated data is updated, an inconsistency is created unless all copies are similarly updated.

Insertion anomalies: It may not be possible to store some information unless some other information is stored as well.

Deletion anomalies: It may not be possible to delete some information without losing some other information as well

14. Illustrate multivalued dependencies and Fourth normal form with example ?

For a table to satisfy the Fourth Normal Form, it should satisfy the following two conditions:

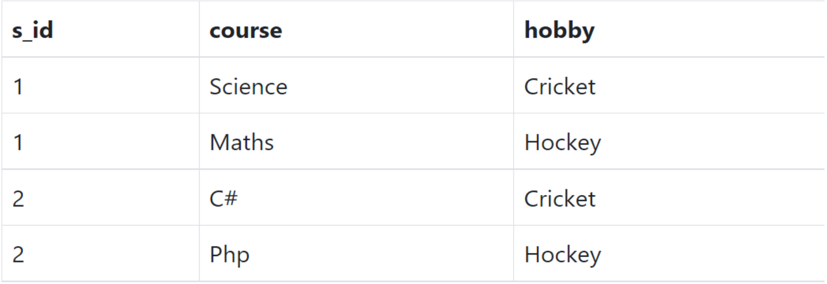
1. It should be in the Boyce-Codd Normal Form.
2. And, the table should not have any Multi-valued Dependency.

A table is said to have multi-valued dependency, if the following conditions are true,

1. For a dependency A → B, if for a single value of A, multiple value of B exists, then the table may have multi-valued dependency.
2. Also, a table should have at-least 3 columns for it to have a multivalued dependency.
3. And, for a relation R(A,B,C), if there is a multi-valued dependency between A and B, then B and C should be independent of each other.

If all these conditions are true for any relation(table), it is said to have multi-valued dependency

Example :

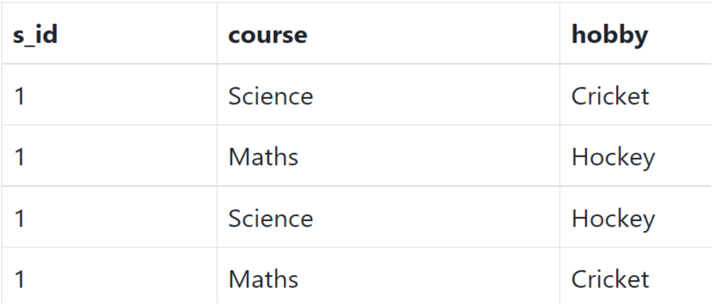


s_id	course	hobby
1	Science	Cricket
1	Maths	Hockey
2	C#	Cricket
2	Php	Hockey

As you can see in the table above, a student with s_id 1 has opted for two courses, Science and Maths, and has two hobbies, Cricket and Hockey.

You must be thinking what problem this can lead to, right?

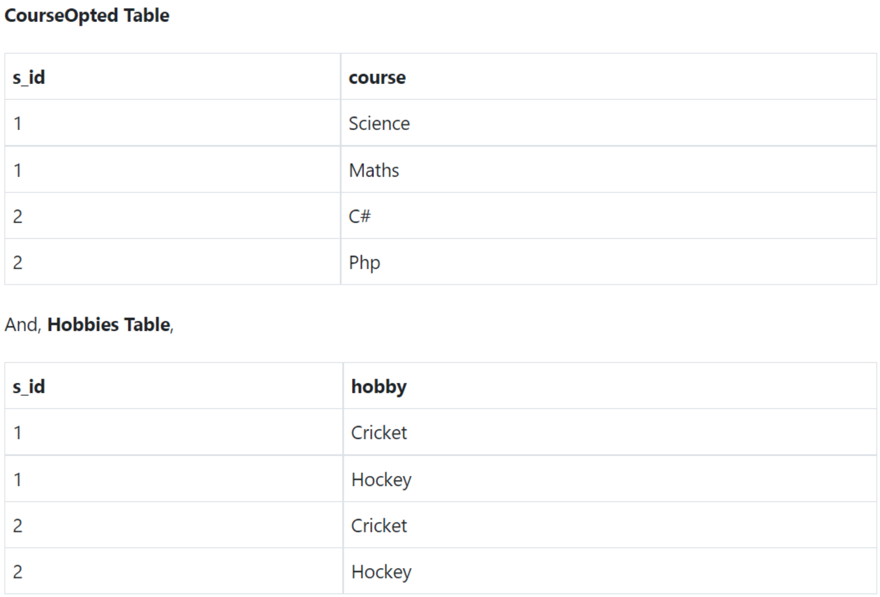
Well the two records for students with s_id 1, will give rise to two more records, as shown below, because for one student, two hobbies exist, hence along with both the courses, these hobbies should be specified.



s_id	course	hobby
1	Science	Cricket
1	Maths	Hockey
1	Science	Hockey
1	Maths	Cricket

And, in the table above, there is no relationship between the columns course and hobby. They are independent of each other.

So there is multivalued dependency, which leads to unnecessary repetition of data and other anomalies as well. To make the above relation satisfy the 4th normal form, we can decompose the table into 2 tables.



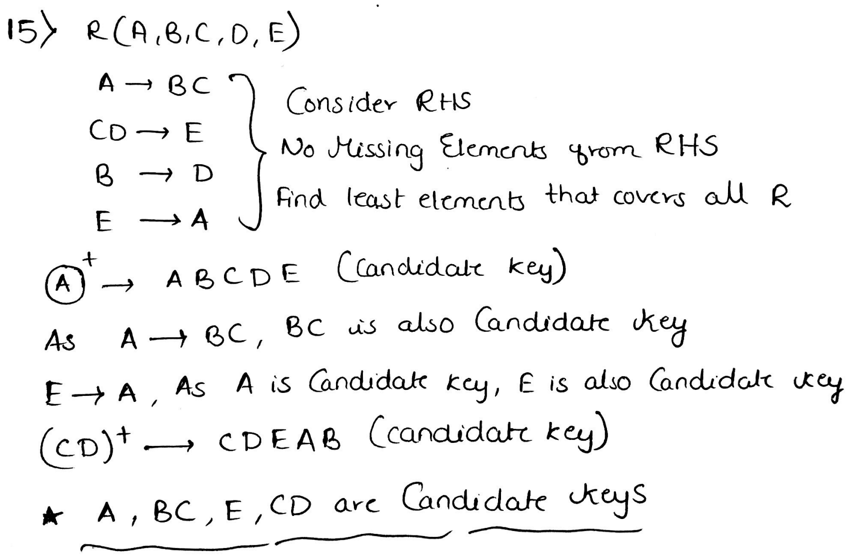
CourseOpted Table

s_id	course
1	Science
1	Maths
2	C#
2	Php

And, **Hobbies Table**,

s_id	hobby
1	Cricket
1	Hockey
2	Cricket
2	Hockey

15. Compute the closure of the following set of functional dependencies for a relation scheme. R(A,B,C,D,E) F= A→BC,CD→E,B→D, E→ A List out the candidate keys of R.



15) R(A,B,C,D,E)

A → BC
CD → E
B → D
E → A

Consider RHS
No Missing Elements from RHS
Find least elements that covers all R

$(A)^+$ → ABCDE (Candidate key)

As A → BC, BC is also Candidate key

E → A, As A is Candidate key, E is also Candidate key

$(CD)^+$ → CDEAB (candidate key)

★ A, BC, E, CD are Candidate keys

16. Write a note on INSERT, DELETE, UPDATE commands in SQL

INSERT , UPDATE , and DELETE are all functions in SQL that help you ensure your data is up-to-date and kept clear of unnecessary or outdated information. INSERT , UPDATE , and DELETE , as well as SELECT and MERGE, are known as Data Manipulation Language (DML) statements, which let SQL users view and manage data.

The INSERT INTO command is used to insert new rows in a table.

Syntax

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

```
INSERT INTO Customers (CustomerName, ContactName, Address, City,
PostalCode, Country)
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger',
'4006', 'Norway');
```

———-

The DELETE statement is used to delete existing records in a table.

```
DELETE FROM table_name WHERE condition;
```

———-

The UPDATE statement is used to modify the existing records in a table.

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

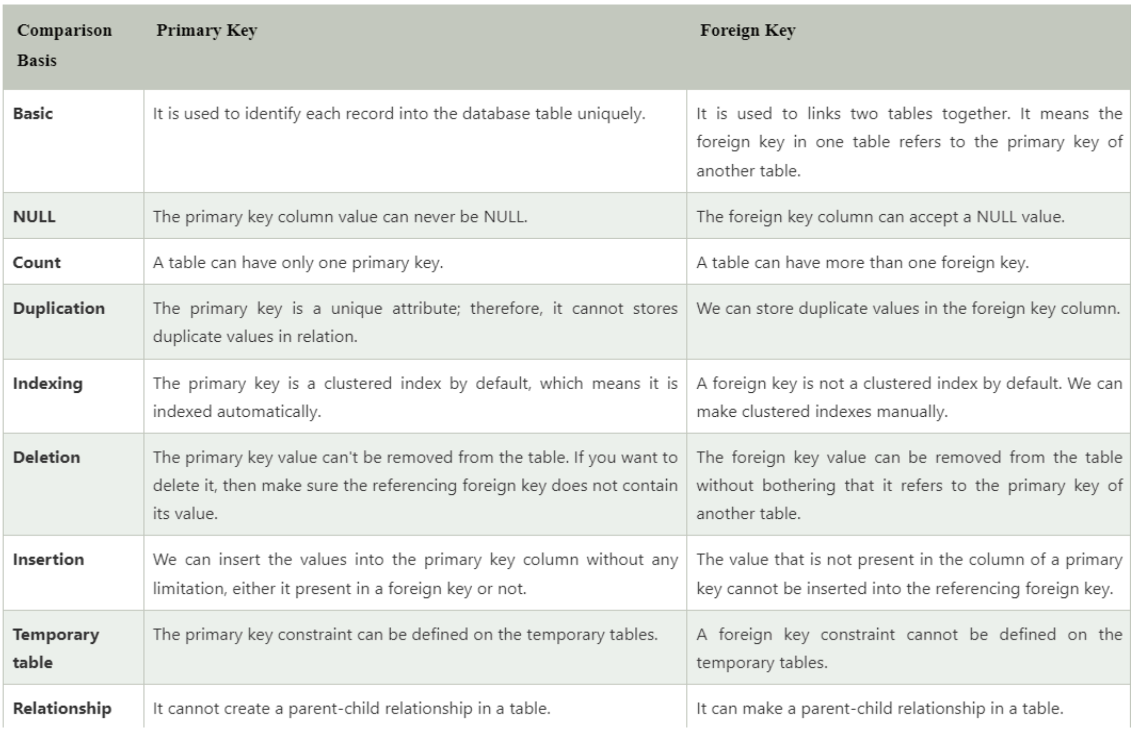
17. R(ABCD) is related to the FD set C→D, C→A, B→C. Find

i. Candidate Key

ii. Normal form that can be existed

iii. Decompose in BCNF relations

18. Explain the key constraints Primary key and Foreign key with examples?



Comparison Basis	Primary Key	Foreign Key
Basic	It is used to identify each record into the database table uniquely.	It is used to links two tables together. It means the foreign key in one table refers to the primary key of another table.
NULL	The primary key column value can never be NULL.	The foreign key column can accept a NULL value.
Count	A table can have only one primary key.	A table can have more than one foreign key.
Duplication	The primary key is a unique attribute; therefore, it cannot stores duplicate values in relation.	We can store duplicate values in the foreign key column.
Indexing	The primary key is a clustered index by default, which means it is indexed automatically.	A foreign key is not a clustered index by default. We can make clustered indexes manually.
Deletion	The primary key value can't be removed from the table. If you want to delete it, then make sure the referencing foreign key does not contain its value.	The foreign key value can be removed from the table without bothering that it refers to the primary key of another table.
Insertion	We can insert the values into the primary key column without any limitation, either it present in a foreign key or not.	The value that is not present in the column of a primary key cannot be inserted into the referencing foreign key.
Temporary table	The primary key constraint can be defined on the temporary tables.	A foreign key constraint cannot be defined on the temporary tables.
Relationship	It cannot create a parent-child relationship in a table.	It can make a parent-child relationship in a table.

19. Find pairs of sids such that the supplier with the first sid charges more for some part than the supplier with the second sid.

$\rho(R1, Catalog)$
$\rho(R2, Catalog)$
$\pi_{R1.sid, R2.sid}(\sigma_{R1.pid=R2.pid \wedge R1.sid \,!=\, R2.sid \wedge R1.cost > R2.cost}(R1 \times R2))$

20. Find the sids of suppliers who supply some red part and some green part

$\rho(R1,\ \pi_{sid}((\pi_{pid}\ \sigma_{color='red'} Parts) \bowtie|\ Catalog))$
$\rho(R2,\ \pi_{sid}((\pi_{pid}\ \sigma_{color='green'} Parts) \bowtie\ Catalog))$
$R1 \cap R2$