BSCCS2001: Graded Assignment with Solutions
Week 1

1. Which of the following is not a drawback of file systems when compared to DBMS?

[MCQ:2 points]

- ◯ Inconsistent data
- √ Ease of initial setup
- ◯ Lack of data integrity
- ◯ Difficult to support concurrency

Solution: The initial setup is more complex for DBMS than a file based system. All the other options are the drawbacks of file systems which a DBMS mitigates.

2. Which of the following creates and maintains the schema of a database?

[MCQ:1 point]

- ◯ Data Manipulation Language
- √ Data Definition Language
- ◯ Data Control Language
- ◯ None of the above

Solution:
Data Definition Language commands are used to define tables, constraints, indexes etc in DBMS. They determine the schema of the database.

3. Which of the following describes the concept that any change made to the physical schema should not affect the logical level of the DBMS?

[MCQ:3 points]

- ◯ Logical Data Independence
- ◯ Logical Data Isolation
- ◯ Physical Data Isolation
- √ Physical Data Independence

Solution:
Physical Data Independence refers to the modification of the physical level without affecting the logical and view level.
Logical Data Independence refers to the modification of the logical level without affecting the view level.

4. Which of the following components of DBMS interacts with the file manager of the operating system?

[MCQ:2 points]

- ◯ Evaluation engine
- ◯ Execution planner
- ◯ Parser
- √ Storage manager

Solution:
Storage manager is responsible for interfacing and monitoring storage access of the DBMS with the operating system.

5. Which of the following is not an example of DBMS?

[MCQ:1 point]

- ◯ Microsoft Access
- ◯ PostgreSQL
- ◯ Sybase
- √ Microsoft Excel

Solution:
Microsoft Excel is a spreadsheet software.

6. Consider the given statements.

- DBMS provides an efficient platform for doing complex arithmetic computation on the data.
- It is easier to create access rules in a file system than in a DBMS.

Choose the correct option.

[MSQ:3 points]

- ◯ Both statements are correct
- √ Both statements are wrong
- ◯ statement 1 is wrong, statement 2 is correct
- ◯ statement 2 is wrong, statement 1 is correct

Solution:
Refer slide 3.15

7. Which type of SQL commands can lead to modification in the Data Dictionary?

[MCQ:3 points]

- √ Data Definition Language.
- ◯ Data Manipulation Language.
- ◯ Dictionary Definition Language.
- ◯ Dictionary Manipulation Language.

Solution: The system modifies the data dictionary whenever a data definition language command is executed.

8. Which component of DBMS maintains the consistency of a database when multiple transactions are executed simultaneously on the data?

[MCQ:2 points]

- ◯ Storage Manager
- ◯ Transaction Management Component
- √ Concurrency Control Manager
- ◯ Query Planner

Solution: Refer slide 05.19

9. Storing multiple copies of the same data within the system is not advisable, because it increases

[MCQ:1 point]

- ◯ Data Consistency
- √ Data Redundancy
- ◯ Atomicity of Data
- ◯ Data Integrity

Solution: Storing multiple copies of same data increases data redundancy. This leads to an inconsistent database when modifications are done on one copy but not done on certain other copies of the same data.

10. Why do we use *try-except* blocks in Python programming language?

[MCQ:2points]

- ◯ For committing data
- ◯ For writing to files
- √ For handling exceptions
- ◯ None of the above

Solution: 'try-except' blocks are used for handling runtime exceptions in Python.

BSCCS2001: Practice/Graded Assignment with Solutions
Week 2

Modules covered:

1. Attribute Types, Relation Schema and Instance, Keys, Relational Query Languages
2. Operations, Select, Project, Union, Difference, Intersection, Cartesian Product
3. Natural Join, Aggregate Operations
4. Introduction to SQL - History of SQL, Data Definition Language (DDL), Basic Query Structure (DML)
5. Additional Basic Operations, Set Operations, Null Values, Aggregate Functions

1. Consider the three relations given below.
   Note that the primary keys are underlined. [MCQ: 1 point]
   employees *(employee_num, employee_name, contact_num, salary)*
   taskAssignment *(employee_num, task_num, task_duration)*
   tasks *(task_num, location)*
   Select the list of possible foreign key(s) for the given relations.

   ◯ *employee_num*

   √ *employee_num, task_num*

   ◯ *task_num*

   ◯ *task_num, task_duration*

Solution: The possible foreign keys are as follows:

- *employee_num* of **taskAssignment** is a foreign key that refers to the relation **employees**.
- *task_num* of **taskAssignment** is a foreign key that refers to the relation **tasks**.

2. Using the table **Students**, choose the correct SQL statement that will return the resultant table given in Figure 2. [MCQ: 2 points]

Name	Age	Country	Score
Tom	13	Australia	70
Lucy	15	Scotland	95
Frank	16	Germany	76
Jane	13	Australia	49
Robert	16	Germany	93
Ryan	18	Ireland	56
Mike	13	Germany	84

Figure 1: Table Students

Age	Country	Count
13	Australia	2
13	Germany	1
15	Scotland	1
16	Germany	2
18	Ireland	1

Figure 2: Resultant table

◯ `SELECT Age, Country, COUNT(*) FROM Students GROUP BY Name;`

◯ `SELECT Age, Country, COUNT(*) FROM Students GROUP BY Age, Score;`

√ `SELECT Age, Country, COUNT(*) FROM Students GROUP BY Age, Country;`

◯ `SELECT Age, Country, COUNT(*) FROM Students GROUP BY Score, Country;`

Solution: The GROUP BY clause is used to group data based on specific values of the given attribute. Here, the tuples are grouped based on attributes Age and Country and and tuples with same values like {13, Australia} are grouped into one tuple. Similarly, tuples with same values like {16, Germany} have also been grouped.

3. Using the table **Students**, choose the correct SQL statement that will return the resultant table given in Figure 4.

[MCQ: 2 points]

Name	Age	Country	Score
Tom	13	Australia	70
Lucy	15	Scotland	95
Frank	16	Germany	76
Jane	13	Australia	49
Robert	16	Germany	93
Ryan	18	Ireland	56
Mike	13	Germany	84

Figure 3: Table Students

Age	Country
18	Ireland
16	Germany
15	Scotland
13	Germany
13	Australia

Figure 4: Resultant table

◯ `SELECT Age, Country FROM Students ORDER BY Score ASC;`

◯ `SELECT DISTINCT Age, Country FROM Students ORDER BY Age ASC;`

◯ `SELECT DISTINCT Age, Country FROM Students ORDER BY Score DESC;`

√ `SELECT DISTINCT Age, Country FROM Students ORDER BY Age DESC;`

Solution: DISTINCT keyword is used to eliminate duplicate records based on the specified attribute(s).
ORDER BY clause is used to sort the data in ascending or descending order, based on one or more columns.
Here, the resultant table will be fetched by retrieving distinct Age and Country, based on sorting the scores in descending order.

Using the table in Figure 5 to answer the questions 4 and 5.

weatherReport

city_code	city	state	temperature	rainfall
1011	Ahmedabad	Gujarat	38	6
1012	Ajmer	Rajasthan	35	4
1013	Aligarh	Uttar Pradesh	37	3
1014	Bengaluru	Karnataka	31	23
1015	Bellary	Karnataka	36	19
1016	Chennai	Tamil Nadu	32	63
1017	Coimbatore	Tamil Nadu	32	40
1018	Hubli	Karnataka	34	26
1019	Jamnagar	Gujarat	34	29
1020	Kota	Rajasthan	37	4

Figure 5: Table **weatherReport**

4. Based on the data given in the table in Figure 5, identify the appropriate query to find the city having minimum rainfall. [MCQ: 3 points]

◯
```
SELECT city
FROM weatherReport
HAVING rainfall = MAX(rainfall);
```

◯
```
SELECT city
FROM weatherReport
WHERE rainfall = MAX(rainfall);
```

◯
```
SELECT t1.city
FROM weatherReport AS t1, weatherReport AS t2
WHERE t1.rainfall < t2.rainfall;
```

√
```
SELECT DISTINCT city
FROM weatherReport
EXCEPT
SELECT DISTINCT t1.city
FROM weatherReport AS t1, weatherReport AS t2
WHERE t1.rainfall > t2.rainfall;
```

Solution: The `HAVING` keyword must be used along with `GROUP BY` keyword. Thus, SQL statement in option 1 is wrong.
The aggregate function like `MAX` must be used in condition with `HAVING` keyword. Thus, SQL statement in option 2 is wrong.

The SQL statement finds out all cities that have rainfall lesser than that of some of the cities. Thus, SQL statement in option 3 is wrong.
The statement:
`SELECT DISTINCT city FROM weatherReport`
selects all the cities.
The statement:
```
SELECT DISTINCT t1.city FROM weatherReport AS t1,
weatherReport AS t2 WHERE t1.rainfall > t2.rainfall;
```
selects all the cities which have rainfall higher than some of the cities. In other words, it extracts all rows except the row with minimum rainfall. The `EXCEPT` keyword returns the rows which are there in the first set of rows, but not there in the second set of rows, i.e. the row that has minimum rainfall. Finally, the SQL statement projects the *city*. Thus, option 4 is correct.
Note: `EXCEPT` is available in the PostgreSQL and SQLite database while `MINUS` is available in MySQL and Oracle.

5. Based on the data given in the table in Figure 5, identify the output for the following SQL statement. [MCQ: 2 points]

```
SELECT city_code
FROM weatherReport
ORDER BY state, city;
```

◯ Output:

city_code
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020

√ Output:

city_code
1011
1019
1015
1014
1018
1012
1020
1016
1017
1013

◯ Output:

city_code
1011
1019
1014
1015
1018
1012
1020
1016
1017
1013

◯ Output:

city_code
1013
1016
1017
1012
1020
1015
1014
1018
1011
1019

Solution: The SQL statement first sorts the table by the *state*, and the output is as follows:

weatherReport

city_code	city	state	temperature	rainfall
1011	Ahmedabad	Gujarat	38	6
1019	Jamnagar	Gujarat	34	29
1014	Bengaluru	Karnataka	31	23
1015	Bellary	Karnataka	36	19
1018	Hubli	Karnataka	34	26
1012	Ajmer	Rajasthan	35	4
1020	Kota	Rajasthan	37	4
1016	Chennai	Tamil Nadu	32	63
1017	Coimbatore	Tamil Nadu	32	40
1013	Aligarh	Uttar Pradesh	37	3

Next, for each *state*, sort the table by *city* and the output is as follows:

weatherReport

city_code	city	state	temperature	rainfall
1011	Ahmedabad	Gujarat	38	6
1019	Jamnagar	Gujarat	34	29
1015	Bellary	Karnataka	36	19
1014	Bengaluru	Karnataka	31	23
1018	Hubli	Karnataka	34	26
1012	Ajmer	Rajasthan	35	4
1020	Kota	Rajasthan	37	4
1016	Chennai	Tamil Nadu	32	63
1017	Coimbatore	Tamil Nadu	32	40
1013	Aligarh	Uttar Pradesh	37	3

Finally, project the column *city_code*, which is option 2.

6. Which of the following statement(s) are **TRUE**? [MSQ: 1 point]

- ◯ All superkeys are candidate keys
- √ All candidate keys are superkeys
- √ A foreign key can be a primary key
- ◯ All superkeys are primary keys

Solution:

- A superkey K is a candidate key if K is minimal. Thus, all candidate keys must be superkeys, but all superkeys need not be candidate keys.
- One of the candidate keys is selected to be the primary key. Thus, the primary key is a candidate key and obviously a superkey. However, minimal superkeys are candidate keys, and one of the candidate keys becomes primary key.
- It is possible that a foreign key can be a primary key.

7. Consider two relations as shown in Figure 6: [MSQ: 3 points]

r1

A	B	C	D
a1	b1	c2	d1
a2	b2	c4	d1
a2	b3	c1	d1
a3	b1	c3	d2
a2	b3	c1	d2
a1	b1	c4	d3
a1	b2	c3	d3

r2

E	B	D
e3	b2	d1
e1	b1	d3
e1	b1	d1
e2	b2	d2
e4	b3	d3
e5	b3	d1

Figure 6: Relations **r1** and **r2**

Identify the correct operation(s) that result(s) in the output shown in Figure 7.

a1	b1	c2	d1	e1
a2	b2	c4	d1	e3
a2	b3	c1	d1	e5
a1	b1	c4	d3	e1

Figure 7: Output tuples

◯ $r1 \times r2$

√ $r1 \bowtie r2$

◯ $\sigma_{r1.B=r2.B \ \wedge \ r1.D=r2.D}(r1 \times r2)$

√ $\pi_{A,r1.B,C,r1.D,E}(\sigma_{r1.B=r2.B \ \wedge \ r1.D=r2.D}(r1 \times r2))$

Solution: Since relations **r1** and **r2** have B and D as common attributes and the given output relation is the set of tuples that have corresponding pairs of B and D values equal in **r1** and **r2**, it follows that the given output is a natural join of **r1** and **r2**. Hence, answer $r1 \bowtie r2$ is correct.
The answer $\pi_{A,r1.B,C,r1.D,E}(\sigma_{r1.B=r2.B \ \wedge \ r1.D=r2.D}(r1 \times r2))$ is also a correct answer, as it is equivalent to $r1 \bowtie r2$.

8. Consider a table **department** that has *salary* as an attribute. What will be the output of the following query?

[MSQ: 1 point]

- SELECT *salary* FROM department WHERE *salary* LIKE '30%5_%_';
  - √ salary with value 305500
  - √ salary with value 305005
  - ◯ salary with value 3050
  - ◯ salary with value 30050

Solution: The percentage sign (%) represents zero, one, or multiple characters and the underscore sign (_) represents a single character.

Use the tables in Figure 8 to answer the questions 9 and 10.

suppliers

sup_num	sup_name
1001	Able
1002	Peter
1003	Molina
1004	Nikki

parts

part_num	sup_num	part_qty
301	1001	32
301	1004	17
301	1002	41
302	1002	11
302	1003	36
302	1001	16
303	1004	25
304	1002	35
304	1003	40

Figure 8: Table **suppliers** and table **parts**

9. Identify the SQL statement(s) that find(s) the names of suppliers who supply parts with *part_num* 301 but do not supply parts with *part_num* 304. [MSQ: 2 points]

◯
```
SELECT sup_name
FROM suppliers s, parts p
WHERE s.sup_num = p.sup_num AND
(part_num = 301 AND part_num <> 304);
```

◯
```
SELECT sup_name
FROM suppliers s, parts p
WHERE s.sup_num = p.sup_num and
(part_num = 301 OR part_num <> 304);
```

√
```
SELECT sup_name
FROM suppliers s, parts p
WHERE s.sup_num = p.sup_num AND part_num = 301
EXCEPT
SELECT sup_name
FROM suppliers s, parts p
WHERE s.sup_num = p.sup_num AND part_num = 304;
```

◯
```
SELECT sup_name
FROM suppliers s, parts p
WHERE s.sup_num = p.sup_num AND part_num = 301
INTERSECT
SELECT sup_name
FROM suppliers s, parts p
WHERE s.sup_num = p.sup_num AND part_num = 304;
```

Solution: From the problem statement: "find the names of suppliers who supply parts with *part_num* 301 but do not supply parts with *part_num* 304" clearly indicates that the desired operation is 'set difference'. In SQL statement, set difference operation is presented by `EXCEPT` keyword. Thus, option 3 is correct.

10. Let $\{sup\_num\}$ be the primary key of the table **suppliers** and $\{part\_num, sup\_num\}$ be the primary key of the table **parts**. [NAT: 3 points]
Consider the SQL query given below:

```
SELECT s.sup_num, sum(p.part_qty)
FROM suppliers s, parts p
WHERE p.part_qty > 30 AND s.sup_num = p.sup_num
GROUP BY s.sup_num
```

How many rows will be returned by the above SQL query?
Answer: 3

Solution: As per the given SQL statement, it first performs a Cartesian product between **suppliers** and **parts**, which output all possible combinations from both the tables.
The part of the statement:
`WHERE p.part_qty > 30 AND s.sup_num = p.sup_num`
eliminates the rows which do not satisfy the condition. The output is as shown below:

s.sup_num	s.sup_name	p.part_num	p.sup_num	p.part_qty
1001	Able	301	1001	32
1002	Peter	301	1002	41
1002	Peter	304	1002	35
1003	Molina	302	1003	36
1003	Molina	304	1003	40

Finally, the part of the statement:
`SELECT s.sup_num, sum(p.part_qty) ...GROUP BY s.sup_num`
results in:

s.sup_num	p.part_qty
1001	32
1002	76
1003	76

Thus, the result has 3 rows.

BSCCS2001: Graded Assignment with Solutions
Week 3

Answer questions 1 and 2 based on the relational schema given in Figure 1.

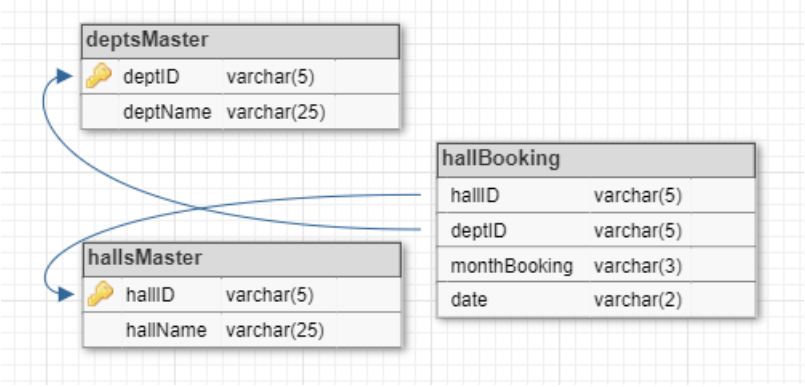


Figure 1: Hall Booking Relational Schema

1. What does the query below return? [MCQ: 2 points]

```
SELECT deptName FROM deptsMaster
WHERE deptID IN (SELECT deptID FROM hallBooking
                 WHERE monthBooking = 'Jan')
    AND deptID NOT in (SELECT deptID FROM hallBooking
                 WHERE monthBooking = 'Feb');
```

- ◯ The names of departments that have booked all halls in the month of January but not in February.
- √ The names of departments that have booked at least one hall in the month of January but never in February.
- ◯ The names of departments that have booked a hall in the month of January or February.
- ◯ The names of departments that have booked at least one hall in the months of January and February.

Solution: The `SELECT` query looks for those departments which has made a booking at least once in January, which is achieved by
`deptID IN (SELECT deptID FROM hallBooking WHERE monthBooking = 'Jan')`
and never in February, which is achieved by
`deptID NOT in (SELECT deptID FROM hallBooking WHERE monthBooking = 'Feb')`.
Since both of these conditions have to be satisfied, these two clauses are connected by `AND`.

2. Find the names of departments that have booked either hall H0001 or hall H0002 or both, in both the months of January and February. [MCQ: 2 points]

◯
```
SELECT deptName FROM deptsMaster
WHERE deptID IN (SELECT deptID FROM hallBooking
                 WHERE hallID IN ('H0001','H0002')
                 AND
                 monthBooking IN ('Jan', 'Feb'));
```

◯
```
SELECT deptName FROM deptsMaster
WHERE deptID IN (SELECT deptID FROM hallBooking
                 WHERE hallID IN ('H0001','H0002')
                 AND deptID IN
                 ((SELECT deptID FROM hallBooking
                 WHERE monthBooking='Jan')
                 INTERSECT
                 (SELECT deptID FROM hallBooking
                 WHERE monthBooking='Feb')));
```

√
```
SELECT deptid FROM hallBooking
WHERE hallID IN ('H0001','H0002') AND monthbooking = 'Jan'
INTERSECT
SELECT deptid FROM hallBooking
WHERE hallID IN ('H0001', 'H0002') AND monthbooking = 'Feb';
```

◯
```
SELECT deptName FROM deptsMaster
WHERE deptID IN (SELECT deptID FROM hallBooking
                 WHERE hallID IN ('H0001','H0002')
                 AND
                 SELECT deptID FROM hallBooking
                 WHERE monthBooking='Jan'
                 INTERSECT
                 SELECT deptID FROM hallBooking
                 WHERE monthBooking='Feb');
```

Solution: `monthBooking IN ('Jan', 'Feb')` will select all the rows that have either January or February as the month of booking.
But, what we need is those departments that have booked in both January and February and NOT either of them.
In Option 2, the first nested query and the second nested query (with two select statements connected by an `INTERSECT` operator) are not looking at rows that satisfy the condition of having booked in both months. They are individually listing out rows that satisfy the two conditions and then the AND is applied. This is insufficient.
In Option 4, the same reasoning as that for Option 2 applies.

In Option 3, it will take common tuples to both -

```
SELECT deptID FROM hallBooking
WHERE hallID IN ('H0001', 'H0002') AND monthBooking='Jan'
```

and

```
SELECT deptID FROM hallBooking
WHERE hallID IN ('H0001', 'H0002') AND monthBooking='Feb'
```

which is the required set of tuples.

3. Consider the relational schema given in Figure 2.

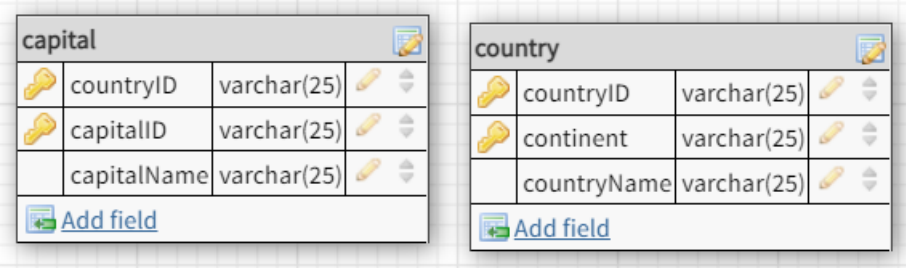


Figure 2: Country Capitals Relational Schema

Choose the SQL statement that returns the capitals of all countries that belong to Asia or Europe. [MSQ: 2 points]

√
```
SELECT capitalName FROM capital
WHERE countryID IN (SELECT countryID FROM country
                    WHERE continent ='Asia'
                    UNION
                    SELECT countryID FROM country
                    WHERE continent ='Europe');
```

◯
```
SELECT capitalName FROM capital
WHERE countryID IN (SELECT countryID FROM country
                    WHERE continent ='Asia'
                    INTERSECT
                    SELECT countryID FROM country
                    WHERE continent ='Europe');
```

◯
```
SELECT capitalName FROM capital
WHERE countryID NOT IN (SELECT countryID FROM country
                    WHERE continent ='Asia'
                    UNION
                    SELECT countryID FROM country
                    WHERE continent ='Europe');
```

√
```
SELECT capitalName FROM capital
WHERE countryID NOT IN (SELECT countryID FROM country
                    WHERE continent != 'Asia'
                    INTERSECT
                    SELECT countryID FROM country
                    WHERE continent != 'Europe');
```

Solution: `countryID IN` filters in and `countryID NOT IN` filters out those countries that satisfy the conditions that follow.

In the first option, two select statements are connected by `UNION` operator in the inner query. One select statement returns countries that belong to Europe, and the second select statement returns countries that belong to Asia. Together with union, the inner query returns all those countries that belong to either Europe or Asia.
In option 2, the connection operator is `INTERSECT`. This will return only those countries which belong to both Asia and Europe.
In option 3, `countryID NOT IN` filters out the countries that belong to either Europe or Asia.
In option 4, `countryID NOT IN` filters out the countries that does not belong to Asia and does not belong to Europe, which effectively filters in those countries that belong to either Europe or Asia.

Use the schema given in Figure 3 to answer the questions 4 to 6.

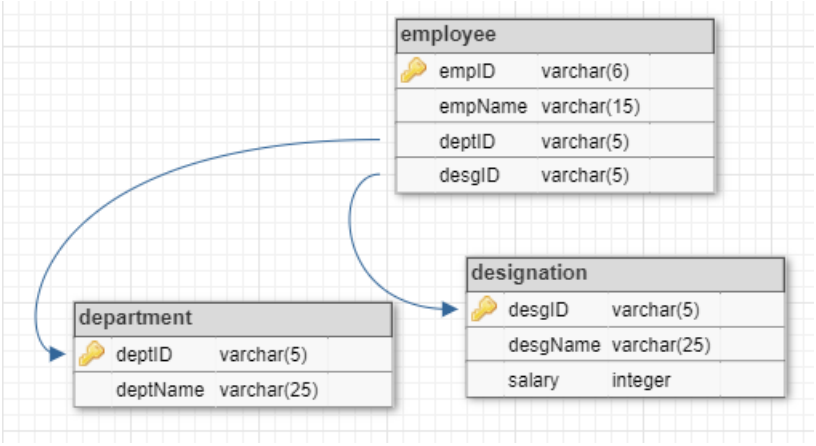


Figure 3: Employee Schema

4. Find the names of employees whose salary is greater than that of every employee in the department 'Sales'. [MCQ: 2 points]

◯
```
SELECT empName FROM employee E NATURAL JOIN designation D
WHERE E.deptID = (SELECT deptID from department, designation
WHERE deptName = 'Sales'
AND salary > ALL (SELECT salary
FROM designation D, department T, employee E
WHERE E.desgID = D.desgID AND T.deptID = E.deptID));
```

◯
```
SELECT empName FROM employee E NATURAL JOIN designation D
WHERE D.salary >= ALL (SELECT salary
FROM designation D, department T, employee E
WHERE E.desgID = D.desgID AND T.deptID = E.deptID
AND T.deptName = 'Sales');
```

◯
```
SELECT empName FROM employee E NATURAL JOIN designation D
WHERE D.salary > SOME (SELECT salary
FROM designation D, department T, employee E
WHERE E.desgID = D.desgID AND T.deptID = E.deptID
AND T.deptName = 'Sales');
```

√
```
SELECT empName FROM employee E NATURAL JOIN designation D
WHERE D.salary > ALL (SELECT salary
FROM designation D, department T, employee E
WHERE E.desgID = D.desgID AND T.deptID = E.deptID
AND T.deptName = 'Sales');
```

Solution: In option 1, the department name is mentioned in the first inner query. So, the salary comparison happens between employees of Sales only.
In option 2, $\geq$ ALL will filter in values that are at least as large as the salary values returned by the inner query.
In option 3, $>$ SOME will filter in all values that are larger than some salary values returned by the inner query.
In option 4, $>$ ALL returns the only value that is larger than all values returned by the inner query. Hence, option 4 is correct.

5. What should be filled in the blanks to find the designation of employees with lowest salary in the Purchase department? [MCQ: 2 points]

```
SELECT DISTINCT desgName FROM designation D1, department T1, employee E1
WHERE E1.desgID = D1.desgID
        AND E1.deptID = T1.deptID
        AND deptName = 'Purchase'
        AND salary ____ (SELECT ____(salary)
                FROM designation D2, department T2, employee E2
                WHERE E2.desgID = D2.desgID
                        AND E2.deptID = T2.deptID
                        AND T2.deptName = 'Purchase');
```

- ◯ $\leq$, MIN
- √ $=$, MIN
- ◯ $<$ ALL, MAX
- ◯ $\leq$ ALL, MAX
- ◯ $\leq$ ALL, MIN

Solution: In order to obtain the least salary within a department, we use $=$ `MIN(salary)` within the department.

6. Based on the relations given in Figure 4, answer the question that follows.

employee

empID	empName	deptID	desgID
E00001	Akash	D0002	G0001
E00002	Akshay	D0002	----(a)----
E00003	Subha	D0003	G0003
E00004	Lavanya	---(b)---	G0002
E00005	Diya	D0001	G0001

department

deptID	deptName
D0001	Purchase
D0002	Sales
D0003	Accounts

designation

desgID	desgName	Salary
G0001	Clerk	5000
G0002	Supervisor	7000
G0003	Manager	10000

Figure 4: Employee instance

What should be filled in blank (b) in the table `employee` in Figure 4, if the query given below returns the value: Lavanya? (Use CAPS for alphabets in the answer) [NAT: 2 points]

```
SELECT empName FROM employee WHERE desgID IN
        (SELECT desgID FROM designation WHERE salary =
                (SELECT MAX(salary)
                FROM employee E, designation G, department D
                WHERE E.deptID = D.deptID
                AND E.desgID = G.desgID
                AND deptName = 'Purchase'))
        AND deptID IN
        (SELECT deptID FROM department WHERE deptName = 'Purchase');
```

Answer: D0001

Solution: The value returned is Lavanya. If we look at the query, it is clear that it is looking for an employee of department 'Purchase' with maximum salary. Clearly, the deptID should be filled by the deptID of the department 'Purchase', which is 'D0001' from the table `department`.

7. Consider the relation **multiple** that stores tuples of the form (a, b) where a is a multiple of b: [MCQ: 2 points]

```
CREATE TABLE multiple(
    first INTEGER,
    second INTEGER,
    PRIMARY KEY (first),
    FOREIGN KEY (second) REFERENCES multiple
            ON DELETE CASCADE);
```

If a tuple (a, b) is deleted from `multiple`, then which of the following is possible?

- ◯ a tuple (c, d) such that c is a factor of a is deleted.
- √ a tuple (c, d) such that c is a multiple of a is deleted.
- ◯ a tuple (c, d) such that d is a factor of a is deleted.
- ◯ a tuple (c, d) such that d is a multiple of a is deleted.
- ◯ a tuple (c, d) such that d is a factor of b is deleted.
- ◯ a tuple (c, d) such that d is a multiple of b is deleted.
- ◯ a tuple (c, d) such that c is a factor of b is deleted.
- ◯ a tuple (c, d) such that c is a multiple of b is deleted.

Solution: The foreign key has been created with `ON DELETE CASCADE`. When a tuple (a, b) is deleted from `multiple`, any tuple of the form (e, a) will be deleted. This leads to any tuple of the form (d, e) to be deleted, and further any tuple of the form (c, d) to be deleted. By the property of being a multiple, observe that a is a multiple of b, e is a multiple of a and b, d is a multiple of e, a and b, and c is a multiple of d, e, a and b and so on. Thus, a tuple (c, d) such that c is a multiple of a is deleted.

8. Consider the relational schema given in Figure 5.

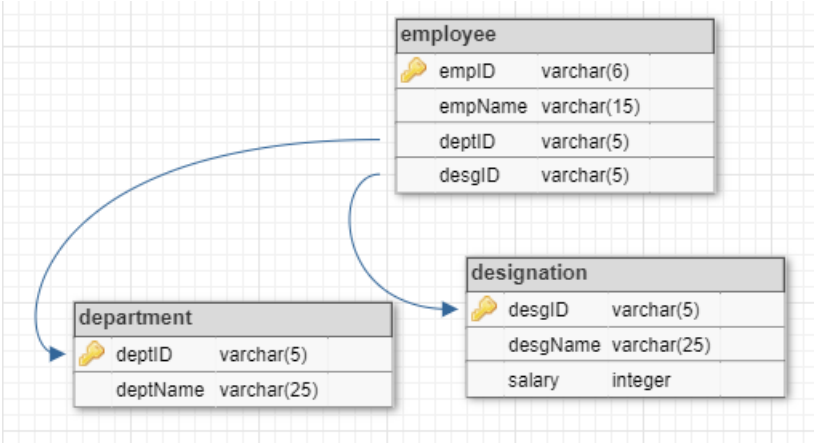


Figure 5: Employee Schema

If the relations **employee, designation** and **department** have 100, 6, 5 rows respectively, what is the maximum number of rows returned by the following query?

[NAT: 2 points]

```
SELECT * FROM employee NATURAL JOIN designation;
```

Answer: 100

Solution: `desgID` is the foreign key in table `employee` that references `designation` table. It follows that the `desgID` in any row of `employee` table will have a corresponding entry in the `designation` table. However the converse is not always true. That is, corresponding to each `desgID` in the `designation` table, there need not be an entry in the `employee` table. Hence there can only be as many rows in the natural join as the number of rows in the `employee` table (since natural join looks for same value of the common attribute). Thus, the maximum number of rows in the natural join in the given example is 100.

9. Consider the table **employee** and table **department** as shown in Figure 6 and answer the question that follows. [MCQ: 2 points]

employee

emp_name	emp_id	age	dept_id
WADE	1	23	10
MADDEN	4	54	10
HARM	6	34	13
TALLY	3	41	16
RODEY	2	46	14
JONES	7	38	14
MULE	5	49	16

department

dept_name	dept_id	dept_location
MATHS	10	Houston
ENGLISH	15	San Antonio
PHYSICS	14	Houston
COMPUTER	13	New York
CHEMISTRY	16	Chicago

Figure 6: employee & department

What will be the output of the following query?

```
SELECT emp_name, dept_name, dept_location FROM employee JOIN department
ON employee.dept_id = department.dept_id
WHERE emp_id IN (6,4,2,7,5) AND age<>54 ORDER BY age asc;
```

◯ Output:

emp_name	dept_name	dept_location
HARM	COMPUTER	New York
TALLY	CHEMISTRY	Chicago
RODEY	PHYSICS	Houston
MULE	CHEMISTRY	Chicago

◯ Output:

emp_name	dept_name	dept_location
HARM	COMPUTER	New York
RODEY	PHYSICS	Houston
MULE	CHEMISTRY	Chicago
JONES	PHYSICS	Houston

◯ Output:

emp_name	dept_name	dept_location
WADE	MATHS	Houston
TALLY	CHEMISTRY	Chicago
RODEY	PHYSICS	Houston
MULE	CHEMISTRY	Chicago

√ Output:

emp_name	dept_name	dept_location
HARM	COMPUTER	New York
JONES	PHYSICS	Houston
RODEY	PHYSICS	Houston
MULE	CHEMISTRY	Chicago

Solution: When table employee and table department are joined on employee.dept_id= department.dept_id, it will fetch the following result -

emp_name	dept_name	dept_location
WADE	MATHS	Houston
MADDEN	MATHS	Houston
HARM	COMPUTER	New York
TALLY	CHEMISTRY	Chicago
RODEY	PHYSICS	Houston
JONES	PHYSICS	Houston
MULE	CHEMISTRY	Chicago

Now as per the query, emp_id should be IN (6,4,2,7,5) AND age must not be 54 and it will fetch the following result -

emp_name	dept_name	dept_location
RODEY	PHYSICS	Houston
MULE	CHEMISTRY	Chicago
HARM	COMPUTER	New York
JONES	PHYSICS	Houston

Now as and when we put ORDER BY age in ascending order, it will fetch the following result -

emp_name	dept_name	dept_location
HARM	COMPUTER	New York
JONES	PHYSICS	Houston
RODEY	PHYSICS	Houston
MULE	CHEMISTRY	Chicago

10. Consider table **Employee**(*eid, edept, ename, esalary, ebonus*) and the code given below.

[MCQ:2 points]

```
CREATE OR REPLACE FUNCTION bonus_fun() RETURNS TRIGGER AS $$
BEGIN
      IF NEW.edept = 'R/D' THEN
             NEW.ebonus = NEW.esalary * .75;
      END IF;
      RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER bonus_trig
    BEFORE INSERT ON Employee
    FOR EACH ROW
    EXECUTE PROCEDURE bonus_fun();

INSERT INTO Employee VALUES (4,'R/D','Diksha',30000);
INSERT INTO Employee VALUES (2,'Accounts','Raj',40000);
SELECT ebonus FROM Employee;
```

What will be the output of the given code?

- ◯ 22500
  0
- √ 22500
  NULL
- ◯ 22500
  30000
- ◯ The code has errors.

Solution: The trigger is executed for the first insert statement since it satisfies the condition described in *bonus_fun*. Accordingly the *ebonus* attribute of the first entry is set to 75% of the *esalary*. For the second insert the trigger condition is not satisfied and hence its *ebonus* is not set therefore it will remain NULL.

BSCCS2001: Graded Assignment with Solutions
Week 4

1. Consider the relations shown in Figure 1. [MSQ: 2 points]

CAR

NAME	MADE	COST
CAR-A	COM-X	200
CAR-B	COM-X	100
CAR-A	COM-Z	300
CAR-B	COM-Y	300
CAR-B	COM-Z	400
CAR-C	COM-X	100
CAR-D	COM-Y	200
CAR-D	COM-X	300

COSTING

MADE	COST
COM-X	100
COM-Z	400

Figure 1: Relations **CAR** and **COSTING**

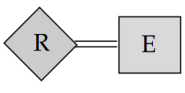
Which car name(s) will be displayed by the operation $CAR \div COSTING$?

- ◯ CAR-A
- √ CAR-B
- ◯ CAR-D
- ◯ CAR-A, CAR-B

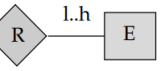
Solution: The relation returned by the division operation must have attributes that are in **CAR** but not in **COSTING**. Thus, the returned relation will have only one attribute *NAME*.
The returned relation must have those tuples from relation **CAR** which are associated to every tuple from **COSTING**. Thus, in this case it will be **CAR-B**.

2. Which of the following symbols is used in the ER-diagrams to represent "total participation of an entity set in a relationship"?

[MCQ: 1 point]

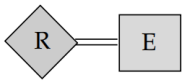
√ 

◯ 

○ 

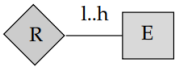
○ None of the above

Solution:



represents total participation of an entity set in a relationship.

represents identifying relationship set for weak entity.



represents cardinality limits.

3. Choose the relational algebra expression that is equivalent to the following tuple calculus expression: [MCQ: 1 point]

$\{t \mid t \in r \wedge (t[A] = 50 \wedge t[B] = 90)\}$

○ $\sigma_{(A=50 \vee B=90)}(r)$

○ $\sigma_{(A=50)}(r) \cup \sigma_{(B=90)}(r)$

√ $\sigma_{(A=50)}(r) \cap \sigma_{(B=90)}(r)$

○ $\sigma_{(A=50)}(r) - \sigma_{(B=90)}(r)$

Solution: Select Operator (σ) selects those rows or tuples from a relation that satisfies the selection condition.
Option 1: It will fetch the tuples having $A = 50$ or $B = 90$.
Option 2: It calculates union of tables having $A = 50$ and $B = 90$ separately.
Option 3: It is valid as it calculates the intersection of tables having $A = 50$ and $B = 90$ separately.
Option 4: The MINUS operator is used to subtract the result set obtained by $\sigma_{(A=50)}(r)$ from the result set obtained by $\sigma_{(B=90)}(r)$, thus it will return only those rows which have tuple $A = 50$ and not those rows which are common to both $A = 50$ and $B = 90$.

4. A bank consists of several **Person** entities. The **Person** entities may have two special types: **Employee** and **AccountHolder**. However, there is a possibility that some **Person** entities are neither an **Employee** nor an **AccountHolder** (like a visitor at the bank). Again, some **Person** entities can be of both **Employee** and **AccountHolder** types. [MCQ: 2 points]
   Identify the constraints on specialization with respect to the above scenario.

   ◯ Disjoint and partial

   √ Overlapping and partial

   ◯ Disjoint and total

   ◯ Overlapping and total

Solution:

- As a **Person** can be an **Employee** or an **AccountHolder** or just a **Person** (like a visitor at the bank), it is partial specialization.
- As a **Person** can be both **Employee** and **AccountHolder**, it is overlapping specialization.

5. Consider the E-R diagram given in Figure 2. [MSQ: 1 point]

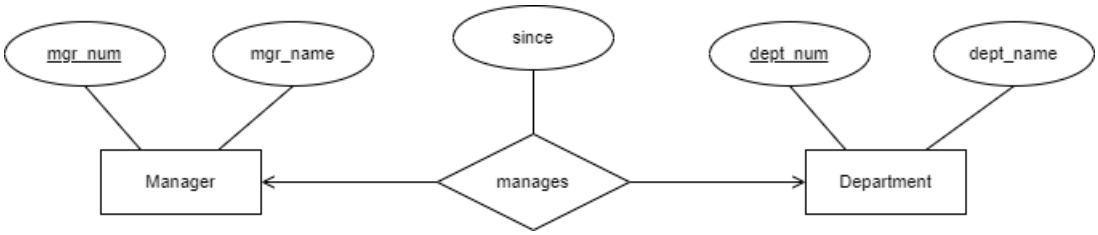


Figure 2: E-R diagram

Identify the option(s) that correctly represent(s) the corresponding tables for the given E-R diagram.

- ○ **Manager**($\underline{mgr\_num}, mgr\_name$)
  manages($\underline{mgr\_num}, \underline{dept\_num}, since$)
  Department($\underline{dept\_num}, dept\_name$)
- √ **Manager**($\underline{mgr\_num}, mgr\_name$)
  Department($\underline{dept\_num}, mgr\_num, dept\_name, since$)
- √ **Manager**($\underline{mgr\_num}, dept\_num, mgr\_name, since$)
  Department($\underline{dept\_num}, dept\_name$)
- ○ **Manager**($\underline{mgr\_num}, \underline{dept\_num}, mgr\_name, since$)
  Department($\underline{dept\_num}, dept\_name$)

Solution: manages is a one-to-one relationship set between **Manager** and **Department**.
The E-R diagram can be mapped to the tables using either of the following:

- **Manager**($\underline{mgr\_num}, mgr\_name$)
- **Department**($\underline{dept\_num}, mgr\_num, dept\_name, since$)

or

- **Manager**($\underline{mgr\_num}, dept\_num, mgr\_name, since$)
- **Department**($\underline{dept\_num}, dept\_name$)

6. Consider the relations below: [MSQ: 3 points]

- **doctor**($\underline{doc\_id}, doc\_name, specialization$)
- **patient**($\underline{patient\_num}, patient\_name$)
- **operationRoster**($\underline{doc\_id, patient\_num}, operation\_cost$)

Identify the appropriate expression(s) to find all the distinct names of the patients operated either by "Dr. Nath" or by "Dr. Joseph".

√ $\prod_{patient\_name}(patient \bowtie \prod_{patient\_num}$
$(\sigma_{doc\_name=\text{“}Dr.\ Nath\text{”}\ \vee\ doc\_name=\text{“}Dr.\ Joseph\text{”}}$
$(doctor \bowtie operationRoster)))$

○ $\prod_{patient\_name}(patient \bowtie \prod_{patient\_num}$
$(\sigma_{doc\_name=\text{“}Dr.\ Nath\text{”}\ \vee\ doc\_name=\text{“}Dr.\ Joseph\text{”}}$
$(doctor \times operationRoster)))$

√ $\prod_{patient\_name}(\sigma_{doc\_name=\text{“}Dr.\ Nath\text{”}}((patient \bowtie (doctor \bowtie operationRoster))))\cup$
$\prod_{patient\_name}(\sigma_{doc\_name=\text{“}Dr.\ Joseph\text{”}}((patient \bowtie (doctor \bowtie operationRoster))))$

○ $\prod_{patient\_name}(patient \bowtie \prod_{patient\_num}$
$(\sigma_{doc\_name=\text{“}Dr.\ Nath\text{”}\ \vee\ doc\_name=\text{“}Dr.\ Joseph\text{”}\ \vee\ doctor.doc\_id=operationRoster.doc\_id}$
$(doctor \times operationRoster)))$

Solution: Option-1 does the following:

1. Apply natural join between **doctor** and **operationRoster**, thus, combines the tuples based on the equality on *doc_id* on both the relations.
2. Then, apply select operation to extract the tuples having *doc_name* as either "Dr. Nath" or "Dr. Joseph".
3. Then, project *patient_num* from the selected tuples.
4. Again, perform natural join between selected *patient_num* tuples with **patient**. Thus, combines the tuples based on the equality on *patient_num* on both the relations.
5. Finally, project the *patient_name*.

Hence, option-1 is **correct**.
In option-2, instead of natural join, Cartesian product has been applied. Since it combines all tuples from **doctor** with all the tuples from **operationRoster**, it is **wrong**.
In option-3, first natural join is applied between **doctor** and **operationRoster** based on equality on *doc_id*. Then, again natural join is applied between the resultant

tuples and **patient** based on equality on *patient_num*. Then, select the tuples having *doc_name* = "*Dr. Nath*" and project the *patient_name*.
The same natural join is again applied between **doctor**, **operationRoster** and **patient**. Then, select the tuples having *doc_name* = "*Dr. Joseph*" and project the corresponding *patient_name*.
Finally, apply union between two sets of tuples. Hence, the option-3 is **correct**.
In option-4, the predicate used for selection is:
doc_name = "*Dr. Nath*" $\vee$ *doc_name* = "*Dr. Joseph*"
$\vee$ *doctor.doc_id* = *operationRoster.doc_id* which is **incorrect**.
The correct form of the predicate is:
(*doc_name* = "*Dr. Nath*" $\vee$ *doc_name* = "*Dr. Joseph*") $\wedge$ (*doctor.doc_id* = *operationRoster.doc_id*).

7. Consider the E-R diagram given in Figure 3.

[MCQ: 2 points]

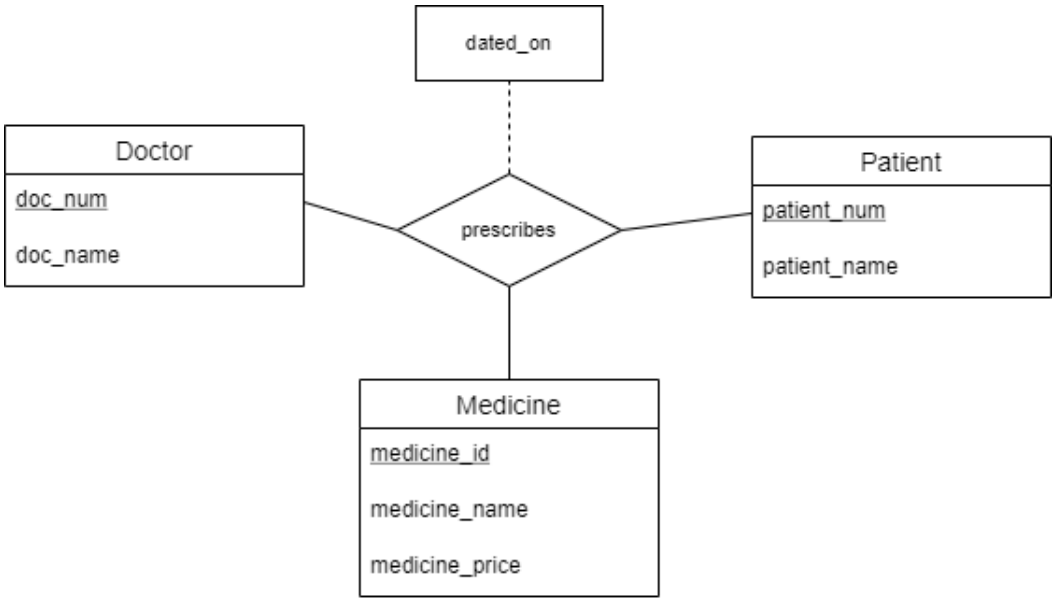


Figure 3: E-R diagram

What will be the schema for the tables corresponding to the relationship set **prescribes**?

- ◯ **prescribes**(*doc_num*, *patient_num*, *medicine_id*)
- ◯ **prescribes**(*doc_num*, *patient_num*, *medicine_id*, *dated_on*)
- ◯ **prescribes**(*doc_num*, *patient_num*, *medicine_id*)

√ **prescribes**($\underline{doc\_num}, \underline{patient\_num}, \underline{medicine\_id}, dated\_on$)

Solution: In the given E-R diagram, there is a ternary relationship with many-to-many relations between the entity sets **Doctor**, **Patient** and **Medicine**. As in the case of binary relationships, the ternary relationship set **prescribes** must also be mapped to a table with attributes as follows:

- the primary keys from all the entity sets associated via the relationship set,
- any descriptive attribute of the relationship set.

Thus, the schema for **prescribes** is:
prescribes($\underline{doc\_num}, \underline{patient\_num}, \underline{medicine\_id}, dated\_on$).

Consider the E-R diagram given in Figure 4 and answer the questions 8 to 10.

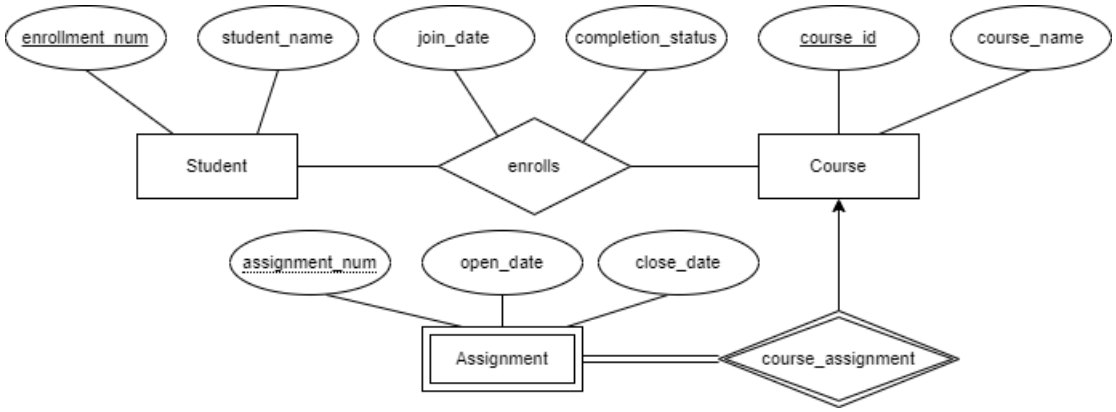


Figure 4: E-R diagram

8. Identify the correct relational schema for the relationship set **enrolls**.

[MCQ: 2 points]

Note: The primary key is underlined.

- ◯ **enrolls** *(join_date, completion_status)*
- ◯ **enrolls** *(enrollment_num, join_date, completion_status)*
- ◯ **enrolls** *(course_id, enrollment_num, join_date, completion_status)*
- √ **enrolls** *(course_id, enrollment_num, join_date, completion_status)*

Solution: As the relationship is many-to-many, the schema for **enrolls** must have primary keys from the associated entity sets and the descriptive attributes of the relationship set. Hence, the right option is:
enrolls *(course_id, enrollment_num, join_date, completion_status)*

9. Identify the correct relational schema for the entity set **Assignment**.

[MCQ: 3 points]

Note: The primary key is underlined.

- ◯ **Assignment** *(assignment_num, open_date, close_date)*
- √ **Assignment** *(course_id, assignment_num, open_date, close_date)*
- ◯ **Assignment** *(assignment_num, course_id, open_date, close_date)*
- ◯ **Assignment** *(assignment_num, course_id, open_date, close_date)*

Solution: Please note that **Assignment** is a weak entity which is identified by the strong entity **Course**. **Assignment** has total participation in the relationship and it is associated with **Course** via **course_assignment** as a many-to-one relationship. Thus, the primary key of **Course** (one-side) entity set will be added to the relational schema for **Assignment** and it also becomes part of the primary key (cannot be null because of total participation). So the schema is:
Assignment *(course_id, assignment_num, open_date, close_date)*

10. With reference to the relationship between **Student** and **Course**, which of the statement(s) is/are **TRUE**?

[MSQ: 3 points]

◯ Each course must have at least one student.

◯ Each student must have enrolled for at least one course.

√ Some courses may have no students.

√ A student may enroll for many courses.

Solution: enrolls is a many-to-many relationship set between **Student** and **Course** entity sets.
As each course may be associated with 0 to n students, option-1 is wrong.
As each student can enroll from 0 to n courses, option-2 is also wrong.
However, option-3 and option-4 are correct.

BSCCS2001: Graded Solutions
Week 5

1. Consider relation **R***(P, Q, C, A, B)* having the following functional dependencies:
   $\mathcal{F} = \{P \to QC, CA \to B, Q \to A, B \to P\}$
   Then, which of the following is correct?

   [MCQ: 2 points]

   ◯ P, B & Q are non-prime attributes.
   ◯ Only P and B are prime attribute.
   ◯ C, A & Q are non-prime attributes.
   √ P, B, Q, C & A are prime attributes.

Solution: Prime Attributes: The attributes that belong to any candidate key are called prime attributes.

Non-prime Attribute: The attributes that do not belong to any candidate key are called non-prime Attributes.

Find out the closure of individual attributes :
$P^+ = P$
$= PQC \ \{P \to QC\}$
$= PQCA \ \{Q \to A\}$
$= PQCAB \ \{CA \to B\}$
$Q^+ = A$
$C^+ = C$
$A^+ = A$
$B^+ = P$
$= PQC \ \{P \to QC\}$
$= PQCA \ \{Q \to A\}$
$= PQCAB \ \{CA \to B\}$

Here, P and B are candidate key, means they are prime attribute .
Now, let us check the combination of Q, C and A
$QC^+ = QC$
$= QCA \ \{Q \to A\}$
$= QCAB \ \{CA \to B\}$
$= QCABP \ \{B \to P\}$

$CA^+ = CA$
$= CAB$ $\{CA \to B\}$
$= CABP$ $\{B \to P\}$
$= CABPQ$ $\{P \to QC\}$
Here, CA and QC are also candidate key. Hence, P, B, Q C and A are prime attributes.

2. Consider relation **Z***(P, Q, R, A, B, C)* having functional dependencies

$\mathcal{F} = \{P \to Q, P \to R, RA \to B, RA \to C, Q \to B\}$
For $PA \to C$ to be the member of $\mathcal{F}^+$, which of the following is true?

[MCQ: 2 points]

- ◯ By augmenting $P \to R$ with A to get $PA \to RA$, and then reflexivity with $RA \to C$.
- ◯ By reflexivity $P \to R$ with A to get $PA \to RA$, and then transitivity with $RA \to C$.
- √ By augmenting $P \to R$ with A to get $PA \to RA$, and then transitivity with $RA \to C$.
- ◯ By reflexivity $P \to R$ with A, to get $PA \to RA$, and then augmenting with $RA \to C$.

Solution:
Armstrong's Axioms:
if $\beta \subseteq \alpha$, then $\alpha \to \beta$ (reflexivity)
if $\alpha \to \beta$, then $\gamma\alpha \to \gamma\beta$ (augmentation)
if $\alpha \to \beta$, and $\beta \to \gamma$, then $\alpha \to \gamma$ (transitivity)

From the given set of FDs
$P \to R$, by augmentation with A, we get $PA \to RA$
also from FDs $RA \to C$, so by transitivity $PA \to C$. Hence, option 3 is correct.

3. The information of all students who have registered for the IIT Madras Online Degree course is given by the relation **studinfo***(studId, name, state)*. The relation **enroll** *(studId, courseId)* gives the list of courses for which each student has enrolled. Let **R** be the relation resulting from the natural join of **studinfo** and **enroll**.
That is, **R** = **studinfo** ⋈ **enroll**.
Then, which of the following statements is/are true? [MSQ:2 points]

- √ The relations **studinfo** and **enroll** result from a lossless decomposition of relation *R*.
- ◯ The relations **studinfo** and **enroll** result from a lossy decomposition of relation *R*.
- √ The number of super keys of *R* is 8.
- ◯ The number of super keys of *R* is 15.

Solution:
studinfo*(studId, name, state)*
enroll *(studeId,courseId)*

R = **studinfo** ⋈ **enroll** = **R***(studId, name, state, courseId)*
Let us check, whether it is lossy or lossless decomposition:
studinfo ∪ **enroll** = *(studId, name, state, courseId)* = *R*
studinfo ∩ **enroll** = *studId* ≠ ϕ, here *studId* is primary key. So, we can determine **studinfo** ∩ **enroll** → **studinfo** or **studinfo** ∩ **enroll** → **enroll**.
Hence, The relation **studinfo** and **enroll** are lossless decomposition of relation **R**.

The number of super keys is given by 2^{n-1}
In relation **R**, the number of attributes is 4 i:e n = 4. So, the number of super keys is 8.

4. A relation $\mathbf{Z}(P,\ Q,\ R,\ S,\ T,\ U,\ V)$ has the following set of functional dependencies: $\mathcal{F} = \{P \rightarrow Q, QR \rightarrow ST, PTU \rightarrow V\}$
What is the closure of the attribute set $\{P, R\}$ under $\mathcal{F}$?

[MCQ: 1 points]

- ◯ $P,\ R,\ S,\ T$
- ◯ $P,\ R,\ U,\ S$
- ◯ $P,\ Q,\ R,\ S,\ T,\ U$
- √ $P,\ R,\ Q,\ S,\ T$

Solution:
$(P, R)^+ = P,\ R$
$= P,\ R,\ Q \quad (P \rightarrow Q)$
$= P,\ R,\ Q,\ S,\ T \quad (QR \rightarrow ST)$
Thus, option 4 is correct.

5. Consider the relation **R***(A, B, C, X, Y, Z)* having the following functional dependencies $\mathcal{F} = \{AB \rightarrow C, C \rightarrow X, X \rightarrow Y, Y \rightarrow Z, Z \rightarrow B\}$
Which of the following is/are true? [MSQ: 3 points]

- √ AC and AY are candidate keys.
- √ All attributes are prime attributes.
- ◯ AC, XY and BC are candidate keys.
- ◯ XY and AZ are candidate keys.

Solution: From the given sets of functional dependencies, if we individually take the closure of *A, B, C, X, Y and Z*, it cannot determine the relation R. Thus, alone *A, B, C, X, Y and Z* can't be a candidate key.
Since there is no incoming arrow to A, A will be the part of some candidate key, but by itself, it is not a candidate key.

Consider the closure of :
$AB^+ = AB$
$= ABC \ \{AB \rightarrow C\}$
$= ABCX \ \{C \rightarrow X\}$
$= ABCXY \ \{X \rightarrow Y\}$
$= ABCXYZ \ \{Y \rightarrow Z\}$

$AC^+ = AC$
$= ACX \ \{C \rightarrow X\}$
$= ACXY \ \{X \rightarrow Y\}$
$= ACXYZ \ \{Y \rightarrow Z\}$
$= ACXYZB \ \{Z \rightarrow B\}$

$AY^+ = AY$
$= AYZ \ \{Y \rightarrow Z\}$
$= AYZB \ \{Z \rightarrow B\}$
$= AYZBC \ \{B \rightarrow C\}$
$= AYZBCX \ \{C \rightarrow X\}$

$AZ^+ = AZ$
$= AZB \ \{Z \rightarrow B\}$
$= AZBC \ \{AB \rightarrow C\}$
$= AZBCX \ \{C \rightarrow X\}$
$= AZBCXY \ \{X \rightarrow Y\}$

Here, *AB, AC, AY and AZ* are candidate keys.

Prime Attributes: Attribute set that belongs to any candidate key are called Prime Attributes.
So, all attributes are prime attributes.

6. Consider a relation **student***(<u>studID</u>, Sname, Age, Sex)* where $studID$ is the primary key. Then, how many super keys are possible for **student**?

[NAT: 1 points]

Ans : 8

Solution:
Consider a relation $R(A_1, A_2, A_3.....A_n)$, a candidate key remaining $A_2, A_3.....A_n$ any subset of attribute which combine with A_1 is a superkey.
Total Keys = 2^{n-1} .
Here, n= 4, So, the number of super keys are 8.

7. Which among the following is a trivial functional dependency?

[MCQ: 1 points]

- ◯ $AB \rightarrow BC$
- ◯ $AB \rightarrow CD$
- ◯ $A \rightarrow B$
- √ $AB \rightarrow B$

Solution: In general, $\alpha \rightarrow \beta$ is trivial if $\beta \subseteq \alpha$. Hence, Option 4 is the right answer.

8. Consider a relation $R(A, B, C, D, E)$ with the following functional dependencies:

$$\mathcal{F} = \{A \to B, A \to D, D \to C, AB \to C, B \to E\}$$

Choose the attribute(s) that are extraneous to any of the functional dependencies in $\mathcal{F}$.

[MSQ: 3 points]

- ◯ A
- √ B
- ◯ C
- ◯ D

Solution: $A \to D, D \to C \Rightarrow A \to C$
It follows that in the FD $AB \to C$, B is extraneous.

9. Given relation $R(A, B, C, D, E)$ and a set of functional dependencies

$$\mathcal{F} = \{A \to B, A \to D, D \to C, AB \to C, B \to E, BD \to CE\}$$

find the prime attribute(s) of R.

[MSQ: 2 points]

- [x] A
- [ ] B
- [ ] C
- [ ] D
- [ ] E

Solution: The attribute that has A in its closure is only A itself. It follows that any candidate key must contain A as a component.
However, since $A^+ = \{ABCDE\}$, it follows that A is a candidate key and hence A is the only prime attribute.

10. Consider a relation $R(A, B, C, D, E)$ having the following functional dependencies:

$$\mathcal{F} = \{A \to BCD, C \to E, B \to D, C \to D, E \to B\}$$

Let $R_1(A, B, C), R_2(A, D, E)$ be a lossless decomposition of R. From among the given options, choose a functional dependency which may be removed from $\mathcal{F}$ that makes the decomposition lossy.

[MCQ: 3 points]

- ◯ $B \to D$
- √ $A \to C$
- ◯ $A \to B$
- ◯ $E \to B$

Solution: In the decomposition $R_1(A, B, C), R_2(A, D, E)$ of R, $R_1 \cap R_2 = A \neq \emptyset$ and $R_1 \cup R_2 = R$ are satisfied.
We have $R_1 \cap R_2 = A$. If A functionally determines either R_1 or R_2, then the decomposition is lossless with respect to $\mathcal{F}$.
We have $A^+/\mathcal{F} = ABCDE$. Hence A is a candidate key and the decomposition is lossless.
Let $\mathcal{F}' = \mathcal{F} \setminus \{A \to C\}$.
$A^+/\mathcal{F}' = ABD$
It follows that $R_1 \cap R_2$ does not functionally determine either R_1 or R_2. Hence the decomposition is lossy with respect to $\mathcal{F}'$.
The decomposition does not become lossy if we remove any other FD. Hence Option 1 is correct.

BSCCS2001: Graded with Solutions
Week 6

1. Consider the relational schema $\mathbf{R}(A, B, C, D, E)$, where the domains of A, B, C, D and E include only atomic values. Identify the possible set of functional dependencies that $\mathbf{R}$ can have such that $\mathbf{R}$ is in 3NF but not in BCNF.

[MCQ: 2 points]

◯ FD: $\{AB \rightarrow CDE\}$

◯ FD: $\{AB \rightarrow CD, B \rightarrow E\}$

◯ FD: $\{AB \rightarrow CD, C \rightarrow D, D \rightarrow E\}$

√ FD: $\{AB \rightarrow CDE, D \rightarrow A, E \rightarrow B\}$

Solution: Given that in $\mathbf{R}$ each attribute is a single-valued attribute. Thus $\mathbf{R}$ is already in 1NF.
Option-1: FD: $\{AB \rightarrow CDE\}$
The only candidate key (thus primary key) is: AB as $(AB)^+ = \{ABCDE\}$.
As all the non-prime attributes are fully functionally dependent on the candidate key, it is already in 2NF.
$\{AB \rightarrow CDE\}$, where AB is a superkey. Thus, it is in 3NF and also in BCNF.
Option-2: $\{AB \rightarrow CD, B \rightarrow E\}$
The only candidate key (thus primary key) is: AB as $(AB)^+ = \{ABCDE\}$.
$B \rightarrow E$ is a partial functional dependency. Thus, it is in 1NF but not in 2NF.
Option-3: FD: $\{AB \rightarrow CD, C \rightarrow D, D \rightarrow E\}$
The only candidate key (thus primary key) is: AB as $(AB)^+ = \{ABCDE\}$.
There is no partial functional dependency. Thus, it is already in 2NF.
$AB \rightarrow CD$, where AB is superkey.
But, for $C \rightarrow D$, $D \rightarrow E$

- the functional dependencies are not trivial.
- L.H.S of the functional dependencies are not superkeys.
- R.H.S of the functional dependencies are not prime attributes.

Thus, these two FDs violate 3NF rules. So, $\mathbf{R}$ is in 2NF but not in 3NF based on this set of FDs.
Option-4: FD: $\{AB \rightarrow CDE, D \rightarrow A, E \rightarrow B\}$
The candidate keys are: AB and DE as $(AB)^+ = \{ABCDE\}$ and $(DE)^+ = \{ABCDE\}$. The prime attributes are A, B, D, E.
There is no partial functional dependency. Thus, it is already in 2NF.
$AB \rightarrow CDE$, where AB is superkey.
For $D \rightarrow A$, $E \rightarrow B$ R.H.S of the functional dependencies are prime attributes. Thus, it is in 3NF. However, these two FDs do not satisfy BCNF (as L.H.S are not superkeys). **So, R is in 3NF but not in BCNF based on this set of FDs**.

2. Consider the relational schema $\mathbf{R}(A, B, C, D, E, F)$, where the domains for A, B, C, D, E and F include atomic values only. If $\mathbf{R}$ satisfies the functional dependencies $\{AB \rightarrow CDE, ABC \rightarrow EF, E \rightarrow F\}$, then identify the correct statement(s).

[MSQ: 2 points]

- ◯ **R** is in 2NF and also in 3NF
- √ **R** is in 2NF but not in 3NF
- ◯ **R** is in BCNF but not in 3NF
- ◯ **R** is in 3NF and also in BCNF

Solution:
Since $(AB)^+ \rightarrow ABCDEF$, AB is the candidate key.
There is no partial functional dependency in set F. Therefore, it is in 2NF.
If we test for 3NF, $E \rightarrow F$ violates 3NF as:

- it is not trivial,
- E is not a superkey,
- F is not prime attribute.

Thus, **R is in 2NF but not in 3NF**

3. Consider the relation $\mathbf{A}(P,Q,R,S,T)$ with the following Functional Dependencies:

   - $PQ \rightarrow RT$
   - $T \rightarrow PQ$
   - $R \rightarrow S$

   What is the highest normal form of the given relation?

   [MCQ: 2 points]

   ◯ 1NF
   √ 2NF
   ◯ 3NF
   ◯ BCNF

Solution: The two possible candidate keys of the given relation are PQ and T. Therefore, the prime attributes are P,Q, and T, and the non-prime attributes are R and S.

Conditions for 2NF

1. It should be in the first normal form.
2. It should not have partial dependencies.

Conditions for 3NF

1. It should be in the second normal form.
2. It should not have transitive dependencies.

The relation is in 2NF because there are no partial dependencies on any of the keys. The relation is not in 3NF because in the functional dependency $R \rightarrow S$, neither R is a superkey nor S is a prime attribute.

4. Consider the relation **IPL**(*TeamID, TeamName, Fours, Sixes*) with the following functional dependencies.

[MCQ: 3 points]

- *TeamID* $\rightarrow$ *(TeamName, Fours, Sixes)*
- *(TeamName, Fours)* $\rightarrow$ *(TeamID, Sixes)*
- *Sixes* $\rightarrow$ *TeamName*

Which one of the following statement(s) is **FALSE** with respect to the information given above?

◯ The relation **IPL** is in 3NF

◯ The functional dependency, *Sixes* $\rightarrow$ *TeamName* violates BCNF

√ The relation **IPL** is in BCNF

◯ All the above

Solution: The relational **IPL** is not in BCNF, because the functional dependency- *Sixes* $\rightarrow$ *TeamName* violates BCNF

5. Consider the following relational schema for the Assignment Evaluation database.
Student($registration\_num, course, enrollment\_num, name, contact\_num, email$)
Assignment($assignment\_num, submission\_date$),
Progress_Report($assignment\_num, registration\_num, grade$).
Given below are the functional dependencies for this schema.

   - $registration\_num \rightarrow course, enrollment\_num, name, contact\_num, email$
   - $course, enrollment\_num \rightarrow name, registration\_num, contact\_num, email$
   - $assignment\_num \rightarrow submission\_date$
   - $assignment\_num, registration\_num \rightarrow grade$

   The given schema is in:

   [MCQ: 3 points]

   ◯ 1NF but not in 2NF
   ◯ 2NF but not in 3NF
   ◯ 3NF but not in BCNF
   √ BCNF

   Solution:
   From the given functional dependencies, the candidate keys of the relations are as follows:

   - $registration\_num$ and $\{course, enrollment\_num\}$ are the candidate keys for the relation **Student**.
   - $assignment\_num$ is the candidate key for the relation **Assignment**.
   - $\{assignment\_num, registration\_num\}$ is the candidate key for the relation **Progress_Report**.

   Thus, all the functional dependencies of the relations in the given schema fulfil the BCNF conditions and the schema is in BCNF.

6. Choose the correct set of option(s):

[MSQ: 1 point]

- ◯ 2NF is considered adequate for relational database design.
- √ A functional dependency of the form A $\rightarrow$ B is trivial if B $\subseteq$ A.
- ◯ A relation produced from an E-R model will always be in BCNF.
- ◯ A functional dependency of the form A $\rightarrow$ B is trivial if A $\subseteq$ B.

Solution: Option 2 follows from the basic definition.

7. Let **P** and **Q** be two relations. Let D(**P**) be a decomposition of **P** based on a set M of functional dependencies. Let D(**Q**) be a decomposition of **Q** based on a set N of functional dependencies. It is known that one among D(**P**) and D(**Q**) is in BCNF and the other is in 3NF.
In order to correctly classify D(**P**) and D(**Q**) as being in BCNF or 3NF, what is the MINIMAL test needed?

[MCQ: 1 point]

- ◯ Test whether both are in BCNF
- √ Test whether one of them is in BCNF
- ◯ Test whether both are in 3NF
- ◯ Test whether one of them is in 3NF

Solution:
Option 1 - Incorrect. This test is enough, but it is not a minimal test.
Option 2 - This is correct. Suppose we test D(**P**) for BCNF. If it is in BCNF, then by the description in the question, we know that D(**Q**) is in 3NF. On the other hand, if it is not in BCNF, then we know that it must be in 3NF and D(**Q**) is in BCNF.
Option 3 - Incorrect. Any decomposition that tests yes for 3NF may or may not be in BCNF. So, we will not be able to classify based on this test alone.
Option 4 - Incorrect. Any decomposition that tests yes for 3NF may or may not be in BCNF. So, we will not be able to classify based on this test alone.

8. Consider the schema $\mathbf{D}(P,\ Q,\ R,\ S)$ with the following functional dependencies
F = $\{R \rightarrow S, P \rightarrow Q, Q \rightarrow R, S \rightarrow P\}$
Let D_1, D_2 be a decomposition of $\mathbf{D}$ such that $D_1 \cap D_2 \neq \phi$.
Then, D_1 and D_2 are

[MCQ: 2 points]

- ◯ not in 2NF
- ◯ in 2NF but not in 3NF
- ◯ in 3NF but not in 2NF
- √ in both 2NF and 3NF

Solution: Candidate keys of relation $\mathbf{D}$ are $P,\ Q,\ R,\ S$.
Decomposition is in both 2NF and 3NF as there is no partial dependency or transitive dependency.

9. Consider the relational schema:
department($dept\_num, dept\_name, mgr\_num, mgr\_name, building\_num, employee\_count, space\_requirement$), where the domains of all the attributes consist of atomic values. Consider the following FDs for the relation **department**.

- $dept\_num \rightarrow mgr\_num, dept\_name$,
- $mgr\_num \rightarrow mgr\_name$,
- $dept\_num, building\_num \rightarrow employee\_count$,
- $employee\_count \rightarrow space\_requirement$,
- $space\_requirement \rightarrow building\_num$

Identify the appropriate decomposition(s) which is/are in BCNF.

[MSQ: 3 points]

√ (mgr_num, mgr_name),
$(dept\_num, mgr\_num, dept\_name)$,
$(employee\_count, space\_requirement)$, and
$(dept\_num, building\_num, employee\_count)$

◯ (mgr_num, mgr_name),
$(dept\_num, mgr\_num, dept\_name)$, and
$(dept\_num, employee\_count, space\_requirement, building\_num)$

◯ (mgr_num, mgr_name),
$(dept\_num, mgr\_num, dept\_name)$,
$(dept\_num, building\_num)$, and
$(building\_num, employee\_count, space\_requirement)$

◯ (mgr_num, mgr_name),
$(dept\_num, mgr\_num, dept\_name)$,
$(space\_requirement, building\_num)$, and
$(dept\_num, employee\_count, space\_requirement)$

Solution:

10. Considering temporal relations, which of the following statement(s) is/are true?

[MSQ: 1 point]

- √ Valid time in a temporal relation is considered as historical information.
- √ Transaction time in a temporal relation is considered as rollback information.
- ◯ Valid time in a temporal relation is considered as rollback information.
- ◯ Transaction time in a temporal relation is considered as historical information.

Solution:

- Valid time provide historical information.
- Transaction time provide rollback information.

BSCCS2001: Graded with Solutions
Week 7

1. Identify the SQL statement(s) that can act as SQL Injections to retrieve all the user IDs, names and passwords from the table *users* in the absence of any security for the database. [MSQ: 2 points]

◯ `SELECT userid, name, password FROM users WHERE userid = 160 or 1<>1;`

◯ `SELECT userid, name, password FROM users WHERE userid IS 160 or 1=1;`

√ `SELECT userid, name, password FROM users WHERE userid = 160 or 1=1;`

√ `SELECT userid, name, password FROM users WHERE userid = 160 or 99=99;`

Solution: SQL injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. It generally allows an attacker to view data that they are not normally able to retrieve. It is one of the most common web hacking techniques.

The SQL Injection statements in option 3 and 4 are valid, since anything preceded by an obvious TRUE statement such as [1=1/99=99] and joined by OR, is always TRUE. And hence the hacker might get access to all the specified data from the *users* table in the absence of any web security for database.

2. Considering the features of web apps, native apps and hybrid apps, select the correct statement(s). [MSQ: 2 points]

◯ Developing a native app is cheaper than developing a hybrid app as it can be built for cross-platforms.

√ Native apps and hybrid apps are available on the App Store and Google Play.

√ Web apps are actually websites that open in your smartphone, PC, tablet, etc. with the help of a web browser.

◯ Native apps are independent of the operating system on which they run.

Solution:

- Hybrid apps have better scalability. Once you have built for one platform, you can launch on another platform with ease. Hence, it is cheaper to develop a hybrid app than a native app.
- Native apps are heavily dependent on the operating system as they are platform specific.

3. From among the following, select the correct statement(s) about sessions and cookies.

    1. A server saves information about cookies it issued, and can use it when serving a request.
    2. Cookies are created and shared between a server and a browser with the help of an HTTP header.
    3. The duration for which a cookie is stored may be permanent.
    4. Sessions are client-side files, whereas cookies are server-side files.

    [MCQ: 2 points]

    √ Only 1, 2 and 3 are correct

    ◯ Only statement 2 and 3 are correct

    ◯ Only statement 1 and 4 are correct

    ◯ All the statements are correct

Solution: Statement 1, 2 and 3 are factual statements about cookies and sessions. Statement 4: Sessions are server-side files whereas, cookies are client-side files.

4. Select the correct statement(s) about embedded SQL: [MSQ: 2 points]

- √ 'EXEC SQL' statement is used to identify embedded SQL request to the pre-processor.
- √ Embedded SQL combines high-level language statements, like those from Java, with SQL.
- ◯ Embedded SQL occurs only in procedures and functions.
- ◯ Embedded SQL are SQL statements that occur only in triggers.

Solution: All are factual statements about embedded SQL.

5. Which of the following layers of a web application can be designed using HTML? [MSQ: 2 points]

- √ Presentation layer
- ◯ Business logic layer
- ◯ Application layer
- ◯ Data access layer

Solution: HTML is used to create a graphical user interface (GUI) to interact with the users, which is a part of presentation layer.

6. Which of the following is not true about an HTTP web server? [MCQ: 2 points]

- ◯ If the document name in a URL identifies an executable program, the HTTP web server executes the program.
- ◯ An HTTP web server accepts requests from several web clients.
- √ An HTTP web server is mainly responsible for creating a graphical user interface.
- ◯ HTTP protocol is used for communication with web server.

Solution: An HTTP web server receives the web requests from web client and the protocol used for the communication is HTTP.
The document name in a URL may identify an executable program, that, when run, generates an HTML document.
However, the graphical user interface is a part of presentation layer. In case of web application, the graphical user interface is provided by the web browser.

7. Which among the following is not the component of three-layer web architecture? [MCQ: 2 points]

- ✓ Graphical User Interface (GUI)
- ◯ Web server
- ◯ Application server
- ◯ Database server

Solution: The three components of three-layer web architecture are: 1) web server, 2) application server, and 3) database server.

8. App development has 4 phases:

    1. data modeling,
    2. process modeling,
    3. modeling, and testing & turnover,
    4. business modeling.

    Identify the correct sequence in which the phases take place. [MCQ: 2 points]

    ◯ 1-2-3-4

    ◯ 4-3-2-1

    ◯ 2-1-4-3

    √ 4-1-2-3

    Solution: App development has 4 phases: business modeling, data modeling, process modeling, and testing & turnover.

9. Consider the following caching facilities:

    1. Caching of JDBC connections between servlet requests
    2. Caching of pages by Web proxy
    3. Caching of generated HTML
    4. Caching results of database queries

    Let A: caching at server site, and B: caching at client network.
    Match the correct caching facility with A and B. [MCQ: 2 points]

    ◯ A:1, 2; B: 3, 4

    ◯ A:1, 2, 3; B: 4

    ◯ A:1; B: 2, 3, 4

    √ A:1, 3, 4; B: 2

    Solution: Please refer to slide: 35.10

10. Which of the following is/are advantages of using Java Server Pages?
[MSQ: 2 points]

- ✓ Java Server Pages are platform independent and portable.
- ◯ Java Server Pages are executed at client side and so, are extremely useful for input validation by avoiding many round trips.
- ◯ Java Server Pages can access client-side resources.
- ✓ Java Server Pages makes it more convenient to write and to modify regular HTML than Servlets, which needs to have a million println statements that generate the HTML.

Solution: Advantages of using Java Server Pages (JSP) are:

- JSP is platform independent and portable.
- JSP is a servlet, but it is more convenient to write and to modify regular HTML than to have a million println statements used in Servlets that generate the HTML.
- Java Server Pages are executed by the web server before the web server sends the HTTP response. It can access server-side resources like databases, catalogues etc.

However, Java Server Pages do not execute at client-end.