1 Database Quiz with Solutions

- Which of the following is the correct order of occurrence in a typical SQL statement?

  □ SELECT, GROUP BY, WHERE, HAVING.
  ■ SELECT, WHERE, GROUP BY, HAVING.
  □ SELECT, WHERE, HAVING, GROUP BY, SELECT, HAVING, WHERE, GROUP BY.
  □ SELECT, HAVING, WHERE, GROUP BY.

  ANSWER: ”WHERE” always comes before ”GROUP BY” and ”HAVING” always comes after ”GROUP BY”.

- Student and Enrolled tables:

SID	name	age	login	GPA
53666	Kayne	A@cs	28	4.0
53655	Tupac	B@cs	26	3.5
53688	Bieber	C@cs	22	3.9

Table 1: STUDENT table.

SID	CID	grade
53666	15 – 415	C
53688	15 – 721	A
53688	15 – 826	B
53655	15 – 415	C
53666	15 – 721	C

Table 2: ENDROLLED table.

- Which of the following is the correct outcome of the SQL query:

```sql
SELECT cid
FROM ENROLLED
WHERE grade = 'C';
```

  ■ Extract the course ids(cid) where student receive the grade C in the course.
  □ Extract the unique course ids(cid) where student receive the grade C in the course.
  □ Error.
  □ None of these.

  ANSWER: The query will extract the course ids where student receive the grade “C” in the course.

- Which of the following is the correct outcome of the SQL query:

```sql
SELECT DISTINCT cid
FROM ENROLLED
WHERE grade = 'C';
```

  □ Extract the course ids where student receive the grade C in the course.
  ■ Extract the Distinct course ids where student receive the grade of C in the course.
  □ Error.
  □ None of these.

  ANSWER: By using DISTINCT keyword you can extract the distinct course ids where student receive the grade of C in the course.

- Which of the following is the correct outcome of the SQL query:

```
SELECT name, cid
FROM STUDENT, ENROLLED
WHERE STUDENT.sid = ENROLLED.sid AND ENROLLED.grade = 'C';
```

  □ Returns the name of all students and their corresponding course ids.

  ■ Returns the name of students and their corresponding course id where they have received grade C.

  □ Error.

  □ None of these.

  ANSWER: The above query will first join the ENROLLED and STUDENT tables, then it will evaluate the WHERE condition and then it will return the name of students and corresponding course id where they received the grade of C.

- Which of the following is the correct outcome of the SQL query:

```
SELECT STUDENT.name, ENROLLED.grade
FROM STUDENT, ENROLLED
WHERE STUDENT.sid = ENROLLED.sid AND ENROLLED.cid = '15-415' AND
      ENROLLED.grade IN ('A','B');
```

  ■ Returns the name, grade of the students who took course ’15-415′ and got a grade’ A’ or ‘B’ in that course.

  □ Returns the name, grade of the students who took the course ’15-415′ but didn’t get grade ‘A’ or ‘B’ in that course.

  □ Error.

  □ None of these.

  ANSWER: The above query will first join the ENROLLED and STUDENT tables, then it will evaluate the WHERE condition and then it will return the name, grade of the students, those took 15-415 and got a grade ‘A’ or ‘B’ in the course. But for the given two tables it will give zero records in output.

- Which of the following query will find all the unique students who have taken more than one course?

  □

```
SELECT DISTINCT e1.sid
FROM ENROLLED AS e1, ENROLLED AS e2
WHERE e1.sid != e2.sid AND e1.cid != e2.cid;
```

  □

```
SELECT DISTINCT e1.sid
FROM ENROLLED AS e1, ENROLLED AS e2
WHERE e1.sid = e2.sid AND e1.cid = e2.cid;
```

  □

```
SELECT DISTINCT e1.sid
FROM ENROLLED AS e1, ENROLLED AS e2
WHERE e1.sid != e2.sid AND e1.cid != e2.cid;
```

  ■

```
SELECT DISTINCT e1.sid
FROM ENROLLED AS e1, ENROLLED AS e2
WHERE e1.sid = e2.sid AND e1.cid != e2.cid;
```

  ANSWER: The last query would be a right option. This query will first apply SELF JOIN on ENROLLED table and then it evaluate the condition (e1.sid = e2.sid AND e1.cid != e2.cid).

- Which of the following statement will add a column *F_name* to the STUDENT table?

  □
  ```
      ALTER TABLE STUDENT ADD COLUMN (’F_name’ VARCHAR(20));
  ```

  ■
  ```
      ALTER TABLE STUDENT ADD ’F_name’ VARCHAR(20);
  ```

  □
  ```
      ALTER TABLE STUDENT ADD (’F_name’ VARCHAR(20));
  ```

  □
  ```
      ALTER TABLE STUDENT ADD COLUMN (’F_name’);
  ```

  ANSWER: ALTER TABLE command allows a user to add a new column to a table. Option 2 is correct syntax of ALTER to add a column in the table.

- Which of the following query(s) will result in a successful insertion of a record in the STUDENT table?

  ```
      INSERT INTO STUDENT (sid,name,login,age,gpa) VALUES (53888, Drake, drake@cs, 29, 3.5);
      INSERT INTO STUDENT VALUES (53888, Drake, drak@ecs, 29, 3.5)
  ```

  ■ Both queries will insert the record successfully.

  □ Query 1 will insert the record successfully and Query 2 will not.

  □ Query 2 will insert the record successfully and Query 1 will not.

  □ Both queries will not be able to insert the record successfully.

  ANSWER: Both queries will successfully insert a row in table STUDENT. The Query 1 is useful when you want to provide target table, columns, and values for new tuples and Query 2 is a Short-hand version of insert command.

- You want to insert a record into the ENROLLED table, which of the following option(s) will insert a row in ENROLLED table successfully?

  ```
      INSERT INTO ENROLLED VALUES(53667, ’15-420’, ’C’);
      INSERT INTO ENROLLED VALUES(53666, ’15-421’, ’C’);
      INSERT INTO ENROLLED VALUES(53667, ’15-415’, ’C’);
      INSERT INTO ENROLLED VALUES(53666, ’15-415’, ’C’);
  ```

  □ 1 and 3.

  □ Only 3.

  ■ 2 and 4.

  □ Only 4.

  ANSWER: ‘Sid’ in “ENROLLED” table is ‘Foreign Key’ referenced by ‘Sid’ in “STUDENT” table, so option 2 and 4 will run successfully because in ENROLLED table’s ‘Sid’ column you can insert those values which are present in STUDENT’s table ‘Sid’ columns due to foreign key.

- Consider the following queries:

  ```
      SELECT name FROM ENROLLED LEFT OUTER JOIN STUDENT  ON STUDENT.sid = ENROLLED.sid;
      SELECT name FROM STUDENT  LEFT OUTER JOIN ENROLLED ON STUDENT.sid = ENROLLED.sid;
  ```

  Which of the following option is correct?

  ■ Queries 1 and 2 will give the same results.

  □ Queries 1 and 2 will give different results.

  □ Query 1 will produce an error and Query 2 will run successfully.

  □ Query 2 will produce an error and Query 1 will run successfully.

  ANSWER: In (LEFT, RIGHT or FULL) OUTER joins, order matters. But both query will give the same results because both are dependent on records present in table and which column in selected.

- Which of the following statements will modify the data type of “Sid” column in ENROLLED table? (There is no Foreign Key relationship between tables STUDENT and ENROLLED).

  ■
  ```
  ALTER TABLE ENROLLED MODIFY (sid VARCHAR(100));
  ```

  □
  ```
  ALTER TABLE ENROLLED MODIFY sid VARCHAR(100);
  ```

  □
  ```
  ALTER TABLE ENROLLED MODIFY COLUMN (sid VARCHAR(100));
  ```

  □
  ```
  ALTER TABLE ENROLLED MODIFY ATTRIBUTE (sid VARCHAR(100));
  ```

  ANSWER: The “ALTER TABLE MODIFY” is used to modify column definition in a table. So option 1 is correct.

- Which of the following statement will remove the ‘Sid’ column from the ENROLLED table? (There is no Foreign Key relationship between tables STUDENT and ENROLLED).

  □
  ```
  ALTER TABLE ENROLLED DROP (sid varchar(10));
  ```

  □
  ```
  ALTER TABLE ENROLLED DROP COLUMN (sid VARCHAR(10));
  ```

  ■
  ```
  ALTER TABLE ENROLLED DROP COLUMN Sid;
  ```

  □
  ```
  ALTER TABLE ENROLLED MODIFY (sid);
  ```

  ANSWER: The “ALTER TABLE DROP COLUMN” can be used to drop a column from the table. So Option 3 is the right answer.

- Which of the following command(s) is / are related to transaction control in SQL?

  □ ROLLBACK
  □ COMMIT
  □ SAVEPOINT
  ■ All of the above.

  ANSWER: All are related to transaction control in SQL.

- Which of the following is true for a Primary Key?

  □ It can take a value more than once.
  □ It can take null values.
  ■ It can’t take null values.
  □ None of these.

  ANSWER: In a relational schema, there exist only one Primary Key and it can’t take NULL values. So option 3 is the correct answer.

- What is the difference between a Primary Key and a Unique Key?

  □ Primary key cannot be a date variable whereas Unique Key can be.
  ■ You can have only one Primary Key whereas you can have multiple Unique Keys.
  □ Primary key can take NULL values but Unique Key cannot NULL values.
  □ None of these.

  ANSWER: You can create a date variable as a Primary Key in table. In relational schema, you can have only one Primary Key and there may be multiple Unique Key present in table. Unique Key can take NULL values.

- Which of the following statement(s) is/are true for UPDATE in SQL?

  1. You can update only a single table using UPDATE command.
  2. You can update multiple tables using UPDATE command.
  3. UPDATE command, you must list what columns to update with their new values (separated by commas).
  4. To update multiple targeted records, you need to specify UPDATE command using the WHERE clause.

  Select the correct option:

  ■ 1, 3 and 4.
  □ 2, 3 and 4.
  □ 3 and 4.
  □ Only 1.

  ANSWER: Options are self-explanatory.

- Which of the following statement is correct about 'CREATE TABLE' command while creating a table?

  ■ We need to assign a data-type to each column.
  □ We have flexibility in SQL so we can assign a data-type to column even after creating a table.
  □ It is mandatory to insert at least a single row while creating a table.
  □ None of these.

  ANSWER: Each column must possess behavioral attributes like data-types and precision in order to build the table structure.

- Which of the following are the synonyms for 'column' and 'row' of a table?

  1. Row = [Tuple, Record]
  2. Column = [Field, Attribute]
  3. Row = [Tuple, Attribute]
  4. Columns = [Field, Record]

  Select the correct option:

  ■ 1 and 2.
  □ 3 and 4.
  □ Only 1.
  □ Only 2.

  ANSWER: In DBMS records are also known as tuple and rows and columns are known as attributes and fields.

- Which of the following operator is used for comparing 'NULL' values in SQL?

  □ Equal
  ■ IS
  □ IN
  □ None of above.

  ANSWER: In SQL if you want to compare a NULL value you need to use IS statement.

- Which of the following statement(s) is/are true about “HAVING” and “WHERE” clause in SQL?

  1. ”WHERE” is always used before “GROUP BY” and HAVING after “GROUP BY”
  2. ”WHERE” is always used after “GROUP BY” and “HAVING” before “GROUP BY”
  3. “WHERE” is used to filter rows but “HAVING” is used to filter groups
  4. “WHERE” is used to filter groups but “HAVING” is used to filter rows

  Select the correct option:

  ■ 1 and 3.

  □ 1 and 4.

  □ 2 and 3.

  □ 2 and 4.

  ANSWER: HAVING is performed after GROUP BY. If you have to apply some conditions to get results, you need to use WHERE before GROUP BY.

- Identify, which of the following column ‘A’ or ‘C’ given in the below table is a ‘Primary Key’ or ‘Foreign Key’? (We have defined ‘Foreign Key’ and ‘Primary Key’ in a single table).

A	C
2	4
3	4
4	3
5	2
7	2
9	5
6	4

  □ Column ‘A’ is Foreign Key and Column ‘C’ is ‘Primary Key’.

  ■ Column ‘C’ is Foreign Key and Column ‘A’ is ‘Primary Key’.

  □ Both can be ‘Primary Key’.

  □ Based on the above table, we cannot tell which column is ‘Primary Key’ and which is ‘Foreign Key’.

  ANSWER: Column A is taking unique values and column A doesn’t have NULL values, so it can be considered as Primary Key of this table. Whereas C is the example of Foreign Key, because all values present in this column are already present in column A.

- What are the tuples additionally deleted to preserve reference integrity when the rows (2,4) are deleted from the below table. Suppose you are using ‘ON DELETE CASCADE’ (use the same table of the question above).

  ■ $(5, 2), (7, 2), (9, 5)$.

  □ $(5, 2), (7, 2)$.

  □ $(5, 2), (7, 2), (9, 5), (3, 4)$.

  □ $(5, 2), (7, 2), (9, 5), (6, 4)$.

  ANSWER: When $(2, 4)$ is deleted, since C is a Foreign Key referring A with DELETE ON CASCADE, all entries with value 2 in C must be deleted. So $(5, 2)$ and $(7, 2)$ are deleted. As a result of this, 5 and 7 are deleted from A, which causes $(9, 5)$ to be deleted.

- Suppose you are given a table/relation “EMPLOYEE” which has two columns (‘Name’ and ‘Salary’). The Salary column in this table has some NULL values. Now, I want to find out the records which have null values.

Name	**Salary**
Saurav	Null
Ankit	1000
Faizan	2000
Sunil	3000
Kunal	4000

  Table 3: EMPLOYEE table.

  What will be the output for the following queries?

  ```
      SELECT *
      FROM EMPLOYEE
      WHERE Salary <> NULL;
  ```

  ```
      SELECT *
      FROM EMPLOYEE
      WHERE Salary = NULL;
  ```

  □ Query 1 will give last 4 rows as output (excluding null value).

  □ Query 2 will give first row as output (only record containing null value).

  ■ Query 1 and Query 2 both will give the same result.

  □ Can’t say.

  ANSWER: If we compare (<>, =) Salary it will give 0 records, below are the following reasons:

  – Salary = Null is unknown.

  – Salary <> Null is unknown.

- What is the difference between TRUNCATE, DELETE and DROP? Which of the following statement(s) is/ are correct?

  1. DELETE operation can be rolled back but TRUNCATE and DROP operations cannot be rolled back.
  2. DELETE operation cannot be rolled back but TRUNCATE and DROP operations can be rolled back.
  3. DELETE is an example of DML (Data Manipulation Language) but remaining are the examples of DDL (Data Definition Language).
  4. All are an example of DDL.

  Select the correct option:

  ■ 1 and 3.

  □ 2 and 3.

  □ 1 and 4.

  □ 2 and 4.

  □ None of the above.

  ANSWER: Options are self-explanatory.

- Tables A, B have three columns (namely: ‘id’, ‘age’, ‘name’) each. These tables have no NULL values and there are 100 records in each of the table. Here are two queries based on these two tables ‘A’ and ‘B’:

```
    SELECT A.id
    FROM A
    WHERE A.age > ALL (SELECT B.age
            FROM B
            WHERE B.name = ’Ankit’);
```

```
    SELECT A.id
    FROM A
    WHERE A.age > ANY (SELECT B.age
            FROM B
            WHERE B.name = ’Ankit’);
```

  Which of the following statement is correct for the output of each query?

  □ The number of tuples in the output of Query 1 will be more than or equal to the output of Query 2.

  □ The number of tuples in the output of Query 1 will be equal to the output of Query 2.

  ■ The number of tuples in the output Query 1 will be less than or equal to the output of Query 2.

  □ Can’t say.

  ANSWER: ANY and ALL operate on subqueries that return multiple values. ANY returns true if any of the subquery values meet the condition. But in case of ALL it will returns the records if all conditions are true. So options 3 is correct.

- Suppose you want to compare three keys (‘Primary Key’, ‘Super Key’ and ‘Candidate Key’) in a database. Which of the following option(s) is/are correct?

  1. Minimal Super Key is a Candidate Key.
  2. Only one Candidate Key can be Primary Key.
  3. All Super Keys can be a Candidate Key.
  4. We cannot find “Primary Key” from “Candidate Key”.

  Select the correct option:

  ■ 1 and 2.

  □ 2 and 3.

  □ 1 and 3.

  □ 2 and 4.

  □ 1,2 and 3.

  ANSWER: Options are self-explanatory

- Consider a relation R with the schema R (A, B, C, D, E, F) with a set of functional dependencies F as follows:

$$F = \{AB \to C, BC \to AD, D \to E, CF \to B\}$$

  Which of the following will be the output of DA^+? (For any X, X^+ is closure of X).

  □ DA

  ■ DAE

  □ ABCD

  □ ABCDEF

  ANSWER: $(DA)^+ = DAE$

- Suppose you have a table “Loan_Records”:

Borrower	Bank_Manager	Loan_Amount
Ramesh	Sunderajan	10000.00
Sumesh	Ramgopal	5000.00
Mamesh	Sunderajan	7000.00

Table 4: LOAN_RECORDS table.

What is the output of the following SQL query:

```
    SELECT Count(*)
    FROM (
       (SELECT Borrower, Bank_Manager FROM Loan_Records) AS S
       NATURAL JOIN
       (SELECT Bank_Manager, Loan_Amount FROM Loan_Records) AS T
      );
```

□ 4

■ 5

□ 8

□ 10

ANSWER:

- Temporary table S is given below:

Borrower	Bank_Manager
Ramesh	Sunderajan
Sumesh	Ramgopal
Mamesh	Sunderajan

- Temporary table T is given below:

Bank Manager	Loan_Amount
Sunderajan	10000.00
Ramgopal	5000.00
Sunderajan	7000.00

- If you apply NATURAL JOIN on both tables (S and T) and evaluate the condition on ‘Bank_Manager’, you will get the following intermediate table after apply:

Borrower	Bank_Manager	Loan_Amount
Ramesh	Sunderajan	10000.00
Ramesh	Sunderajan	7000.00
Sumesh	Ramgopal	5000.00
Mamesh	Sunderajan	10000.00
Mamesh	Sunderajan	7000.00

- “Sunderjan” appears two times in Bank_Manager column, so their will be four entries with Bank_Manager as “Sunderjan”. So count(*) will give the 5 output in OUTER query.

- Is “SELECT” operation in SQL equivalent to “PROJECT” operation in relational algebra?

  □ Yes, both are equivalent in all the cases.

  ■ No, both are not equivalent.

  ANSWER: In relational algebra ‘PROJECT’ operation gives the unique record but in case of ‘SELECT’ operation in SQL you need to use DISTINCT keyword for getting unique records.

- IV1 table:

Name	Salary	Company	Designation
Saurav	1000	AV1	Junior Data Scientist
Ankit	800	AV1	Data Scientist
Sunil	1200	AV2	Senior Manager
Kanal	1400	AV2	CEO
Deepak	1100	AV3	Data Entry Operator
Swati	1200	AV3	BDE
Falzan	900	AV1	Deep Learning Expert

Table 5: AV1 table.

- What will be the output of following query:

```
    SELECT Name FROM AV1 WHERE Name LIKE ’\%a\%’;
```

  □ Saurav, Ankit, Kunal, Deepak, Swati, Faizan.

  ■ Saurav, Kunal, Deepak, Swati, Faizan.

  □ Kunal, Deepak, Swati, Faizan.

  □ None of above.

  ANSWER: The query will search for records in column ‘Name’ which will have at least one ‘a’ and LIKE operation is case sensitive, so option 1 will not be the answer; Hence option 2 is true.

- What will be the output for the below query:

```
    SELECT Name FROM AV1 WHERE Name LIKE ’\%______%’;
```

  ■ It will return names where number of characters in names are greater than or equals to 6.

  □ It will return names where number of characters in names are greater than 6.

  □ It will return names where number of characters in names are less than or equals to 6.

  □ It will give an error.

  ANSWER: The query will search for records in column ‘Name’ where the number of characters in names are greater than or equal to 6.

- Which of the following is true for TRUNCATE in SQL?

  □ It is usually slower than DELETE command.

  ■ It is usually faster than DELETE command.

  □ There is no comparison between DELETE and TRUNCATE.

  □ TRUNCATE command can be rolled back.

  ANSWER: TRUNCATE is faster than delete because TRUNCATE is a DDL command so it does not produce any rollback information and the storage space is released while the DELETE command is a DML command and it produces rollback information too and space is not deallocated using DELETE command.

- What will be the output of the below query:

```sql
SELECT Company, AVG(Salary)
FROM AV1
HAVING AVG(Salary) > 1200
GROUP BY Company
WHERE Salary > 1000;
```

□

Company	AVG
AV2	1300

□

Company	AVG
AV3	1150
AV2	1300

□

Company	AVG
AV3	1200
AV2	1300

■ None of these.

ANSWER: This query will give the error because 'WHERE' is always evaluated before 'GROUP BY' and 'HAVING' is always evaluated after 'GROUP BY'.

- What will be the output for the query:

```sql
SELECT MAX(Salary)
FROM AV1
WHERE Salary < ( SELECT MAX(Salary)
      FROM AV1
      );

```

```sql
WITH S AS (SELECT Salary, ROW_NUMBER() OVER(ORDER BY Salary DESC) AS RowNum
       FROM AV1
      )
SELECT Salary
FROM S
WHERE RowNum = 2;
```

■ Query 1 output = 1200 and Query 2 output = 1200

□ Query 1 output = 1200 and Query 2 output = 1400

□ Query 1 output = 1400 and Query 2 output = 1200

□ Query 1 output = 1400 and Query 2 output = 1400

ANSWER: Both queries will generate the second-highest salary in AV1 which is 1200; Hence option 1 is true.

- Consider the following relational schema:

```
Students (rollno: INTEGER, sname: STRING)
Courses (courseno: INTEGER, cname: STRING)
Registration (rollno: INTEGER, courseno: INTEGER, percent: REAL)
```

  Which of the following query would be able to find the unique names of all students having score more than 90% in the courseno 107?

  ■
```
SELECT DISTINCT S.sname
FROM Students AS S, Registration AS R
WHERE R.rollno = S.rollno AND R.courseno = 107 AND R.percent > 90;
```

  □
```
SELECT UNIQUE S.sname
FROM Students AS S, Registration AS R
WHERE R.rollno = S.rollno AND R.courseno = 107 AND R.percent > 90;
```

  □
```
SELECT sname
FROM Students AS S, Registration AS R
WHERE R.rollno = S.rollno AND R.courseno = 107 AND R.percent > 90;
```

  □ None of these.

  ANSWER: Option 1 is true. Option 2 will give the error ("UNIQUE" is not used in SQL) and in option 3 unique names will not be the output.

- Consider the relation T1 (A, B) in which (A, B) is the Primary Key and the relation T2 (A, C) where A is the Primary Key. Assume there are no NULL values and no Foreign Keys or Integrity Constraints. Which of the following option is correct related to following queries?

```
SELECT A
FROM T1
WHERE A IN (SELECT A FROM T2 );
```

```
SELECT A
FROM T2
WHERE A IN (SELECT A FROM T1 );
```

  □ Both queries will definitely give the same result.

  ■ Both queries may give the same result.

  □ Both queries will definitely give a different result.

  □ None of these.

  ANSWER: For the same values (values should be unique) for the column A in tables T1 and T2. Query 1 and Query 2 will give the same output, hence option 2 is true.

- Which of the following option is correct about following queries?

```
SELECT emp.id, department.id
FROM emp NATURAL JOIN department;
```

```
SELECT emp.id, department.id
FROM department NATURAL JOIN emp;
```

  ■ Both queries will give same outputs.

  □ Both queries will give different output.

  □ Need table structure.

  □ None of these.

  ANSWER: For NATURAL JOINS, the order doesn't matter. The queries will return same results.

- Indexing is useful in a database for fast searching. Generally, B-tree is used for indexing in a database. Now, you want to use Binary Search Tree instead of B-tree. Suppose there are numbers between 1 and 100 and you want to search the number 35 using Binary Search Tree algorithm. Which of the following sequences CANNOT be the sequence for the numbers examined?

  □ 10, 75, 64, 43, 60, 57, 55

  □ 90, 12, 68, 34, 62, 45, 55

  ■ 9, 85, 47, 68, 43, 57, 55

  □ 79, 14, 72, 56, 16, 53, 55

  ANSWER: In BST on right side of parent number should be greater than it, but in C after 47, 43 appears that is wrong.

- If an index scan is replaced by sequential scan in SQL, then what will happen? (Number of observations is equal to 1 million).

  □ Execution will be faster.

  ■ Execution will be slower.

  □ Execution will not be affected.

  □ None of these.

  ANSWER: The addition of the index made the query execution faster since the sequential scan is replaced by the index scan.

- Suppose you have a table 'Employee'. In Employee table, you have a column named Salary. Now, you applied Query 1 on Employee table:

  ```
      SELECT *
      FROM Employee
      WHERE (Salary * 100) > 5000;
  ```

  After that, you created an index on Salary columns and then you re-run the Query 2 (same as Query 1):

  ```
      SELECT *
      FROM Employee
      WHERE (Salary * 100) > 5000;
  ```

  Here, Query 1 is taking T1 time and Query 2 is taking T2 time. Which of the following is true for the queries time?

  □ T1 > T2

  □ T2 > T1

  ■ T1 ∼ T2

  □ Can't say.

  ANSWER: The addition of the index didn't change the query execution plan. The index on rating will not work for the query (Salary∗100 > 5000). Theoretically it might work in this case, but obviously the system is not "smart" enough to work that way; But you can create an index on (Salary∗100) which will help.

- What is true for ‘View’ in SQL?

  1. It can help in providing security.
  2. It can be used for hiding complexity.
  3. If there are more than one table involved in the view, we cannot perform (Data Manipulation Language) DML queries.
  4. When you drop the base table, view becomes inactive.

  Select the correct option:

  □ 1 and 3.

  □ 2 and 4.

  □ 1, 3 and 4.

  ■ All of these.

  ANSWER: All options are correct.

- Suppose I created a table called ”Avian” using below SQL query:

```
CREATE TABLE Avian (Emp_id SERIAL PRIMARY KEY, Name VARCHAR);
```

  Now, I want to insert some records in the table Avian:

```
INSERT INTO Avian (Name) VALUES("FRAZY");
INSERT INTO Avian (Name) VALUES("ANKIT");
INSERT INTO Avian (Name) VALUES("SUNIL");
INSERT INTO Avian (Name) VALUES("SAURAV");
```

  Which of the following will be the output of the query:

```
SELECT * FROM Avian;
```

■

Emp_id	Name
1	FRAZY
2	ANKIT
3	SUNIL
4	SAURAV

□

Emp_id	Name
	FRAZY
	ANKIT
	SUNIL
	SAURAV

□ Error.

□ None of these.

ANSWER: At the time of table creation Avian, we have used SERIAL for ”Emp_id” which autoincrement ”Emp_id” whenever you insert a record in table Avian; Hence option 1 is true.