

DBT MCQ Test - 21

Aug18/ DBT/M156

Database Technologies

Diploma in Advance Computing

August 2018

Date: **­­­­­­­­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

PRN: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name: ­­­­­­­­­­­­­­­­­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Note: Attempt all questions. Each question carries 1 mark. No Negative Marking.***

1. The HAVING clause is used to restrict the results returned by the GROUP BY clause.

1. **True**
2. False

2. The GROUP BY Clause is used to group rows with same values.

1. **True**
2. False

3. The GROUP BY clause must appear after the FROM and \_\_\_\_\_\_\_ clauses.

1. **WHERE**
2. HAVING
3. ORDER BY
4. LIMIT

4. MySQL allows you to sort the groups in ascending or descending orders.

1. **True**
2. False

5. The default order is........... in Group By clause.

1. **Ascending**
2. Descending
3. Invalid
4. None of the above

6. If the GROUP BY clause is omitted, the HAVING clause behaves like the \_\_\_\_\_ clause.

1. **WHERE**
2. ORDER BY
3. LIMIT
4. None of the above

7. The HAVING clause applies a filter condition to each \_\_\_\_\_\_\_\_\_\_.

1. First 10 rows
2. **group rows**
3. individual row
4. None of the above

8. The WHERE clause applies the filter condition to each \_\_\_\_\_\_\_\_\_\_.

1. First 10 rows
2. group rows
3. **individual row**
4. None of the above

9. What are the results of the following queries if col is an integer column?

1. SELECT \* FROM mytbl WHERE num\_col = '4';

2. SELECT \* FROM mytbl WHERE num\_col = 4;

**a) Same**

b) Different

c) 1 is an error

d) 2 is an error

10. What is the meaning of "EMPTY SET"?

**a) No values**

b) Error

c) Access denied

d) None of the mentioned

11. Select odd one out?

a) Equality Conditions

b) Inequality Conditions

c) Range condition

**d) Between**

12. The following query belongs to which condition types?

SELECT fname FROM person WHERE title = ’TELLER’;

**a) Equality condition**

b) Inequality condition

c) Range condition

d) All of the mentioned

13. The following query belongs to which condition types?

SELECT fname FROM person WHERE fed\_id=’111-11-111’;

**a) Equality condition**

b) Inequality condition

c) Range condition

d) All of the mentioned

14. The following query belongs to which condition types?

SELECT firstName FROM person WHERE Dept\_id = (SELECT Dept\_id FROM department WHERE locationID = 1001);

/\* where locationID is a unique key \*/

**a) Equality condition**

b) Inequality condition

c) Range condition

d) All of the mentioned

15. Is the following query belongs to the “Equality condition”?

SELECT product\_type.name, product.name FROM product\_type INNER JOIN Product ON product\_type.dept = Product.dept WHERE product\_type.name = ’customers\_accounts’;

**a) Yes**

b) No

c) Depends

d) None of the mentioned

16. What is the meaning of “Equality Conditions”?

**a) Equal to**

b) Not equal to

c) Both Equal to and Not equal to

d) None of the mentioned

17. What is the meaning of “Inequality Condition”?

**a) Not-equal-to**

b) Equal-to

c) Both Not-equal-to and Equal-to

d) None of the mentioned

18. Does the following query belong to the “Inequality condition”?

SELECT product\_type.name, product.name FROM product\_type INNER JOIN Product ON product\_type.dept=Product.dept WHERE product\_type.name <> ’customers\_accounts’;

**a) Yes**

b) No

c) Depends

d) None of the mentioned

19. Does the following query belong to the “Inequality condition”?

SELECT product\_type.name, product.name FROM product\_type INNER JOIN Product ON product\_type.dept=Product.dept WHERE product\_type.name ! = ’customers\_accounts’;

**a) Yes**

b) No

c) Depends

d) None of the mentioned

20. What is the meaning of “Range Conditions”?

a) Expression is equal to Expression

b) Expression is not equal to Expression

**c) Expression fall under certain range**

d) None of the mentioned

21. Which among the following data types can be used with “Range Condition”?

a) Numeric data type

b) Temporal data type

c) String data type

**d) Both Numeric and Temporal data type**

22. The following query belongs to which “Condition Types”?

SELECT emp\_id, fname, lname FROM account WHERE start\_date < ’2007-10-08’;

a) Equality conditions

b) Inequality condition

**c) Range condition**

d) None of the mentioned

23. What will be the output of the following query?

SELECT \* FROM employee WHERE start\_date BETWEEN ‘2007-01-01’ AND ‘2008-01-01’;

**a) All employees details between 2007 and 2008**

b) All employees details before 2008

c) All employees details from 2007 to 2008

d) None of the mentioned

24. Fill the blanks with suitable options?

BETWEEN \_\_\_\_\_\_ AND \_\_\_\_\_\_

a) Upper and lower limit

**b) Lower and upper limit**

c) Both a and b

d) None of the mentioned

25. What will be the output of the following query?

SELECT \* FROM employee WHERE start\_date >= ’2007-01-01’ AND Start\_date <= ’2005-01-01’

a) All employees between 2007 and 2005

b) All employees from 2007 to 2005

c) Empty set

**d) None of the mentioned**

26. What will be the output of the following query?

SELECT customer\_id, product\_id, avail\_balance FROM account WHERE avail\_balance BETWEEN 3000 AND 5000.

**a) Only those details will be shown whose as available balance form 3000 to 5000**

b) Only those details will be shown whose available balance is more than 3000

c) Only those details will be shown whose available balance is less than 5000

d) None of the mentioned

27. Which among the following operators is/are belongs to “Inequality conditions”?

a) <>

b) !=

c) =

**d) Both <> and !=**

28. Which among the following operators is/are belongs to “Equality conditions”?

a) <>

b) !=

**c) =**

d) >/<

29. Which among the following operators is/are belongs to “Range conditions”?

a) <>

b) !=

c) =

**d) >/<**

30. Find the error in the following query?

SELECT cust\_id, fed\_id FROM customer WHERE cust\_id = ’I’ AND fed\_id BETWEEN 5000-00-000 AND 9999-999-000;

a) Yes

**b) No**

c) Range too high to compare

d) None of the mentioned

31. What will be the output of the following query?

SELECT account\_id, product\_id, cust\_id FROM account WHERE product\_id IN (‘sav’, ‘chd’, ‘mm’);

**a) Only those values are selected whose product\_id is either ‘sav’, ‘chd’, ‘mm’**

b) Only those values are selected whose product\_id is either ‘sav’’

c) Only those values are selected whose product\_id is either ‘sav’, ‘chd’

d) All of the mentioned

32. What will be the output of the following query?

SELECT account\_id, product\_id, cust\_id FROM account WHERE product\_id NOT IN (‘sav’, ‘chd’, ‘mm’);

**a) Only those values are selected whose product\_id does not belong to either ‘sav’, ‘chd’, ‘mm’**

b) Only those values are selected whose product\_id does not belong to either ‘sav’’

c) Only those values are selected whose product\_id does not belong to either ‘sav’, ‘chd’

d) All of the mentioned

33. Consider a database table COLORS name whose attributes are ID (primary key), Color.

ID = {1, 2, 3, 4, 5, 6}

Color = {Red, Blue, Red, Blue, Green, Yellow}

What will be the output of the following query?

SELECT ID FROM COLORS WHERE Color IN (Green, Yellow);

**a) {5, 6}**

b) {1, 2, 3}

c) {3, 4}

d) None of the mentioned

34. Consider a database name COLORS whose attributes are ID (primary key), Color.

ID = {1, 2, 3, 4, 5, 6}

Color = {Red, Blue, Red, Blue, Green, Yellow}

What will be the output of the following query?

SELECT ID FROM COLORS WHERE Color not IN (Green, Yellow);

a) {5, 6}

b) {1, 2, 3}

c) {3, 4}

**d) None of the mentioned**

35. Consider a database name COLORS whose attributes are ID (primary key), Color.

ID = {1, 2, 3, 4, 5, 6}

Color = {Red, Blue, Red, Blue, Green, Yellow}

What will be the output of the following query?

SELECT ID FROM COLORS WHERE Color NOT IN (Green, Yellow, Blue);

a) {5, 6}

b) {1, 2, 3}

c) {3, 4}

**d) {1, 3}**

36. Consider a database name COLORS whose attributes are ID (primary key), Color.

ID = {1, 2, 3, 4, 5, 6}

Color = {Red, Blue, Red, Blue, Green, Yellow}

What will be the output of the following query?

SELECT ID FROM COLORS WHERE color IN (Green, Yellow, Blue);

a) {5, 6}

b) {1, 2, 3}

c) {3, 4}

**d) {2, 4, 5, 6}**

37. Consider a database name COLORS whose attributes are ID (primary key), Color.

ID = {1, 2, 3, 4, 5, 6}

Color = {Red, Blue, Red, Blue, Green, Yellow}

What will be the output of the following query?

SELECT ID FROM COLORS WHERE color IN (Green, Yellow, Blue, Red);

a) {5, 6}

b) {1, 2, 3}

**c) {1, 2, 3, 4, 5, 6}**

d) Empty set

38. Consider a database name COLORS whose attributes are ID (primary key), Color.

ID = {1, 2, 3, 4, 5, 6}

Color = {Red, Blue, Red, Blue, Green, Yellow}

What will be the output of the following query?

SELECT ID FROM COLORS WHERE color NOT IN (Green, Yellow, Blue, Red);

a) {5, 6}

b) {1, 2, 3}

c) {1, 2, 3, 4, 5, 6}

**d) Empty set**

39. What will be the output of the following query?

SELECT emp\_id, fname, lname FROM employee WHERE LEFT (lname, 1) = ’T’;

**a) Only those employees are selected whose last name started with ‘T’**

b) Only those employees are selected whose last name started with other than ‘T’

c) All of the mentioned

d) None of the mentioned

40. What will be the output of the following query?

SELECT emp\_id, fname, lname FROM employee WHERE LEFT (lname, 1) =’F’;

**a) Only those employees are selected whose last name started with ‘F’**

b) Only those employees are selected whose last name started with other than ‘F’

c) All of the mentioned

d) None of the mentioned

41. What will be the output of the following query?

SELECT emp\_id, fname, lname FROM employee WHERE LEFT (fname, 1) =’F’;

**a) Only those employees are selected whose first name started with ‘F’**

b) Only those employees are selected whose first name started with other than ‘F’

c) All of the mentioned

d) None of the mentioned

42. What will be the output of the following query?

SELECT emp\_id, fname, lname FROM employee WHERE LEFT (fname, 1) = ’F’ OR LEFT (lname, 1) = ’F’;

**a) Only those employees are selected whose first name and last name started with ‘F’**

b) Only those employees are selected whose first name started with ‘F’ but last name can be starts with any other letter

c) Only those employees are selected whose first name and last name started with any other letter except ‘F’

d) None of the mentioned

43. What will be the output of the following query?

SELECT \* FROM employee WHERE lname LIKE ‘\_a%e%’;

a) All employees whose last name start with any letter but second letter should be ‘a’

b) All employees whose last name start with any letter but contain at least one ‘e’ in his name

c) All employees whose last name should have letter ‘a’ in second position and at least one ‘e’ in his name

**d) All of the mentioned**

44. What will be the output of the following query?

SELECT \* FROM employee WHERE lname LIKE ‘F%’ OR lname LIKE ‘%T’;

a) All employees whose last name should started with ‘F’

b) All employees whose last name end with ‘T’

**c) All employees whose last name should started with ‘F’ and end with ‘T’**

d) None of the mentioned

45. What will be the output of the following query?

SELECT \* FROM employee WHERE lname LIKE ‘F%’ AND lname LIKE ‘%T’;

a) All employees whose last name should started with ‘F’ and end with ‘T’

b) All employees whose last name end with ‘T’ and start with ‘F’

**c) Both a and b**

d) None of the mentioned

46. What will be the output of the following query?

SELECT \* FROM employee WHERE lname LIKE %bas% OR fname LIKE %bbs%;

a) All employees whose last name should contain substring “bas”

b) All employees whose first name should contain substring “bbs”

**c) Both a and b**

d) None of the mentioned

47. Consider an EMPLOYEE relation whose attributes are ID (primary key), firstName, lastName, credit\_card, and salary.

EMPLOYEE = {ID int, firstName varchar (12), lastName varchar (12) credit\_card boolean, salary float (8, 2)}

What will be the output of the following query?

SELECT \* FROM E WHERE CREDIT\_CARD = FALSE;

1. All employee who have credit card
2. **All employee who doesn’t have credit card**
3. All of the mentioned
4. None of the mentioned

48. Consider an EMPLOYEE relation whose attributes are ID (primary key), firstName, lastName, credit\_card, and salary.

EMPLOYEE = {ID int, firstName varchar (12), lastName varchar (12) credit\_card boolean, salary float (8, 2)}

What will be the output of the following query?

SELECT \* FROM E WHERE CREDIT\_CARD = 0;

1. All employee who have credit card
2. **All employee who doesn’t have credit card**
3. All of the mentioned
4. None of the mentioned

49. Consider an EMPLOYEE relation whose attributes are ID (primary key), firstName, lastName, credit\_card, and salary.

EMPLOYEE = {ID int, firstName varchar (12), lastName varchar (12) credit\_card boolean, salary float (8, 2)}

What will be the output of the following query?

SELECT \* FROM E WHERE CREDIT\_CARD = true;

1. **All employee who have credit card**
2. All employee who doesn’t have credit card
3. All of the mentioned
4. None of the mentioned

50. Consider an EMPLOYEE relation whose attributes are ID (primary key), firstName, lastName, credit\_card, and salary.

EMPLOYEE = {ID int, firstName varchar (12), lastName varchar (12) credit\_card boolean, salary float (8, 2)}

What will be the output of the following query?

SELECT \* FROM E WHERE CREDIT\_CARD = 1;

1. **All employee who have credit card**
2. All employee who doesn’t have credit card
3. All of the mentioned
4. None of the mentioned

51. Consider an EMPLOYEE relation whose attributes are ID (primary key), firstName, lastName, credit\_card, and salary.

EMPLOYEE = {ID int, firstName varchar (12), lastName varchar (12) credit\_card boolean, salary float (8, 2)}

What will be the output of the following query?

SELECT \* FROM E WHERE CREDIT\_CARD = 2-2;

1. All employee who have credit card
2. **All employee who doesn’t have credit card**
3. Error in the statement
4. None of the mentioned

52. Consider an EMPLOYEE relation whose attributes are ID (primary key), firstName, lastName, credit\_card, and salary.

EMPLOYEE = {ID int, firstName varchar (12), lastName varchar (12) credit\_card boolean, salary float (8, 2)}

What will be the output of the following query?

SELECT \* FROM E WHERE CREDIT\_CARD = 2-1;

1. **All employee who have credit card**
2. All employee who doesn’t have credit card
3. Error in the statement
4. None of the mentioned

53. Which one is correct syntax for Where clause in SQL server?

a) SELECT WHERE "Condition" Col1, Col2 FROM "Table" ;

b) SELECT "Condition" Col1, Col2 FROM "Table" WHERE;

**c) SELECT Col1, Col2 FROM "Table" WHERE "condition";**

d) None of the above

54. What is the purpose of Order By Clause in SQL server?

**a) It is used to sort the result.**

b) It is used to change sequence order of columns

c) It can’ be used in SQL Server

d) None of the above

55. Order by can only be used by Where Clause, correct?

a) True

**b) False**

56. What needs to be added when user want to show results by Descending Order?

a) Descending order cannot be possible.

**b) User can add DESC with Order By clause**

c) User can add ‘<>ASC’ with Order by Clause.

d) None of the above

57. What is the default order of Order by Clause?

1. Descending
2. **Ascending**
3. Random
4. None of the above

58. Among the below Order By queries, which are correct ones?

1. SELECT \* FROM Table Order By Column;
2. SELECT \* FROM Table Order By Column ASC;
3. SELECT \* FROM Table Order By Column DESC;
4. **All of the above**

59. Aggregate function SUM() cannot be given in……..

1. Order by clause
2. Group by clause
3. **Where clause**
4. Both B and C

60. Aggregate function MAX() be given in……..

1. **Having by clause**
2. Group by clause
3. Where clause
4. All of the above.