PROJECTION

---

1. WQTD EMPLOYEE FIRST NAME AND STATUS FROM EMPS TABLE?

A: SELECT FNAME, STATUS FROM EMPS;

2. WQTD EMPLOYEE LNAME, SALARY AND JOB FROM EMPS TABLE?

A. SELECT LANME, SAL, JOB FROM EMPS;

3. WQTD EMPLOEE COMMISSION, DATE OF BIRTH AND DATE OF JOINING FROMS EMPS TABLE?

A. SELECT COMM, DOB, DOJ FROM EMPS;

4. WQTD DETAILS OF EMPS FROM EMPS TABLE?

A. SELECT \* FROM TABLE;

5. WQTD EMPLOYEE FIRST NAME, LAST NAME AND ANNUAL SALARY FROM EMPS TABLE?

A. SELECT FNAME, LNAME, SAL\*12 FROM EMPS;

EXPRESSION

---

6. WQTD EMPLOYEE FNAME, JOB AND SALARY WITH 20000 BONUS?

A. SELECT FNAME, JOB, SAL+20000 FROM EMPS;

7. WQTD DETAILS OF EMPLOYEE ALONG WITH ANNUAL COMMISSION?

A. SELECT \*, COMM\*12 FROM EMPS;

8. WQTD EMPLOYEE FNAME, LID AND SALARY WITH 10% HIKE?

A. SELECT FNAME, LID, SAL+ (SAL\*10/100) FROM EMPS;

9. WQTD EMPLOYEE FNAME, LNAME, JOB, SAL WITH 15% HIKE AND COMM WITH 5% DEDUCTION?

A. SELECT FNAME, LNAME, JOB, SAL+(SAL\*15/100), COMM-(COMM\*5/100) FROM EMPS;

10. WQTD EMPLOYEE FNAME, JOB AND ANNUAL SAL WITH 10% HIKE?

A SELECT FNAME, JOB, (SAL+(SAL\*10/100))\*12 FROM EMPS;

11. WQTD EMPLOYEE FNAME, JOB, COMM WITH 4% HIKE AND ANNUAL SAL WITH 8% DEDUCTION?

A. SELECT FNAME, JOB, COMM+(COMM\*4/100), (SAL-(SAL\*8/100))\*12 FROM EMPS;

ALIASING

---

IT IS USED TO PROVIDE ALTERNATIVE NAME FOR A COLUMN IN RESULTANT TABLE.

RULE

---

1. WITH OR WITHOUT USING AS KEYWORD WE CAN WRITE ALIAS NAME.

2. WE CAN USE MULTIPLE WORDS AS A ALIAS NAME BY CONNECTING IT WITH UNDER\_SCORE OR BY PASSING INSIDE THE QUOTES.

SELECT SAL AS SALARY

FROM EMPS; :VALID

SELECT SAL SALARY

FROM EMPS; :VALID

SELECT SAL\*12 AS 'ANNUAL SALARY'

FROM EMPS; :VALID

SELECT SAL\*12 AS ANNUAL SALARY

FROM EMPS; :NOT VALID

SELECT SAL\*12 ANNUAL\_SALARY

FROM EMPS; :VALID

---

12. WQTD FNAME AS FIRST NAME, LNAME AS LAST NAME AND ANNUAL SALARY?

A. SELECT FNAME AS 'FIRST NAME', LANME AS 'LAST NAME', SAL\*12 AS 'ANNUAL SALARY' FROM EMPS;

13. WQTD UNIQUE JOB ROLES FROM EMPS??

A. SELECT

DISTINCT

---

IT IS USED TO AVOID DUPLICATE VALUES FROM RESULTANT TABLE.

RULES:

---

1. EITHER \* OR DISTINCT MUST BE THE FIRST ARGUMENT IN SELECT CLAUSE.

2. WE CAN USE MULTIPLE COLUMNS ALONG WITH DISTINCT KEYWORD, IT WILL CHECK FOR COMBINATION OF DUPLICATE VALUES.

---

14. WQTD UNIQUE COMBINATION OF JOB AND SALARY?

A. SELECT DISTINCT JOB,SAL FROM EMPS;

SELECTION

---

IT IS USED TO RETRIVE THE DATA FROM THE TABLE BY SELECTING COLUMN\_NAME OR EXPRESSION AND BY PROVING CONDITIONS.

ORDER OF EXECUTION

---

1. FROM

2. WHERE

3. SELECT

15. WQTD DETAILS OF EMPS IF EMPLOYEE FNAME IS DIVYA?

A. SELECT \* FROM EMPS WHERE FNAME='DIVYA';

16. WQTD FNAME AS FIRST NAME , LNAME AS LAST NAME ALONG WITH JOB IF EMPS ARE WORKING AS WAITER?

A. SELECT FNAME AS 'FIRST NAME', LNAME AS 'LAST NAME', JOB FROM EMPS WHERE JOB='WAITER';

17. WQTD DETAILS OF EMPLOYEES ALONG WITH ANNUAL SALARY IF EMPS SAL>45000?

A. SELECT \* , SAL\*12 AS 'ANNUAL SALARY' FROM EMPS WHERE SAL>45000;

18. WQTD FNAME, LNAME, JOB AND DOB IF EMPS BORN AFTER THE YEAR 1990?

A. SELECT FNAME, LNAME, JOB, DOB FROM EMPS WHERE DOB>'1990-12-31';

19. WQTD FNAME, JOB, DOJ IF EMPLOYEE HIRED BEFORE THE YEAR 2019?

A. SELECT FNAME, JOB, DOJ FROM EMPS WHERE DOJ < '2019-01-01'; OR (WHERE DOJ <= '2019-12-31')

20. WQTD DETAILS OF EMPS WHO WERE

A. SELECT \* FROM EMPS WHERE DOJ > '2020-03-31';

21. EMPLOYEE, FNAME, JOB, ANNUAL SALARY IF EMPLOYEE ANNUAL SALARY IS MORE THAN 500000?

A. SELECT FNAME,JOB, SAL\*12 'ANNAUL SALARY' FROM EMPS WHERE (SAL\*12) >500000;

OPERATORS

---

1. ARITHMATIC OPERATOR(+, -, \*, /, %)

2. RELATIONAL OPERATOR(<, >, <=, >=, =, !=)

3. LOGICAL OPERATOR (AND, OR, NOT)

4. SPECIAL OPERATORS (IN, NOT IN, IS, BETWEEN, NOT BETWEEN, LIKE, NOT LIKE)

5. SUBQUERY OPERATORS (ALL, ANY)

22. WQTD DETAILS OF EMPS IF THE EMPS ARE WORING AS CHEF OR CASHIER?

A. SELECT \* FROM EMPS WHERE JOB='CHEF' OR JOB = 'CASHIER';

23. WQTD FNAME, LNAME, SAL, DOB FROM EMPS TABLE IF EMPS BORN DURING THE YEAR 1995?

A. SELECT FNAME, LNAME, SAL, DOB FROM EMPS WHERE DOB >= '1995-01-01' AND DOB <= '1995-12-31';

24. WQTD DETAILS OF THE EMPLYOESS WHO ARE WORKING AS WAITER AND GETTING SALARY MORE THAN 50000?

A. SELECT \* FROM EMPS WHERE JOB='WAITER' AND SAL>50000;

25. WQTD DETAILS OF EMPS IF THE EMPS ARE WORKING AS SECURITY, CHEF, WAITER OR DELIVERY AND GETTING SAL MORE THAN 45000 AND STATUS OS OFFLINE?

A. SELECT \* FROM EMPS WHERE (JOB='SECURITY' OR JOB='CHEF' OR JOB='WAITER' OR JOB='DELIVERY') AND (SAL>45000) AND (STATUS='OFFLINE');

IN

--

IT IS A MULTI VALUE OPERATOR WHICH TAKES MULTIPLE VALUES AT THE RHS AND SINGLE VALUE AT THE LHS.

SYYNTAX:

  LHS RHS

COLUMN\_NAME/EXPRESSION IN (V1, V2,.....VN);

>IT WORKS ON OR CONDITION.

26. WQTD DETAILS OF EMPS IF EMPS ARE WPRKING AS CHEF, MANAGER OR WAITER AND THERE STATUS IS EITHER BUSY OR AVAILABLE AND JOINED IN THE YEAR 2010 BUT NOT AS FEMALE?

A. SELECT \* FROM EMPS WHERE JOB IN ('CHEF', 'MANAGER', 'WAITER') AND STATUS IN('AVAILABLE', 'BUSY') AND (DOJ >= '2010-01-01' AND DOJ <= '2010-12-31') AND (GENDER != 'F');

27.WQTD DETAILS OF EMPS THOSE WHO ARE NOT WORKING AS CHEF,WAITER,MANAGER(WITHOUT USING SPECIAL OPERATOR)???

SELECT \*

FROM EMPS

WHERE JOB!='CHEF' AND JOB!='WAITER' AND JOB!='MANAGER';

NOT IN:

------

IT IS A MULTI VALUE OPERATOR WHICH TAKES MULTIPLE VALUES AT THE RHS AND SINGLE VALUE AT THE LHS.

SYNTAX:

LHS RHS

COLUMN\_NAME/EXPRESSION NOT IN(V1,V2.......,VN)

>IT WORKS ON AND CONDITION.

SELECT \*

FROM EMPS

WHERE JOB!='CHEF' AND JOB!='WAITER' AND JOB!='MANAGER';

SELECT \*

FROM EMPS

WHERE JOB NOT IN('CHEF','WAITER','MANAGER');

28.WQTD DETAILS OF EMPS IF EMPS ARE NOT GETTING SALARY AS 26000,50000,100000 AND HIRED IN THE YEAR 2016 AND STATUS IS AVAILABLE???

SELECT \*

FROM EMPS

WHERE SAL NOT IN(26000,50000,100000) AND DOJ>='2016-01-01' AND DOJ<='2016-12-31' AND STATUS='AVAILABLE';

29.WQTD FNAME AND SAL IF EMPLOYEES ARE GETTING SALARY MORE THAN OR EQUAL TO 35000 AND LESS THAN OR EQUAL TO 50000???

SELECT FNAME,SAL

FROM EMPS

WHERE SAL>=35000 AND SAL<=50000;

BETWEEN

-------

WHENEVER WE NEED TO INCLUDE SOME RANGE OF VALUES WE USE BETWEEN.

SYNTAX:

>= <=

COLUMN\_NAME/EXPRESSION BETWEEN LOWER\_RANGE\_VALUE AND HIGHER\_RANGE\_VALUE;

SELECT FNAME,SAL

FROM EMPS

WHERE SAL BETWEEN 35000 AND 50000;

30.WQTD DETAILS OF EMPS THOSE WHO WERE BORN IN THE YEAR 1995???

SELECT \*

FROM EMPS

WHERE DOB BETWEEN '1995-01-01' AND '1995-12-31';

31.WQTD DETAILS OF EMPS WHO ARE GETTING SALARY MORE THAN 35000 AND LESS THAN 100000 AND WORKING AS WAITER,CLEANER OR CASHIER??

SELECT \*

FROM EMPS

WHERE SAL>35000 AND SAL<100000 AND JOB IN('WAITER','CLEANER','CASHIER');

OR

SELECT \*

FROM EMPS

WHERE SAL BETWEEN 35000+0.01 AND 100000-0.01 AND JOB IN('WAITER','CLEANER','CASHIER');

32.WQTD DETAILS OF EMPS WHO ARE NOT GETTING SALARY IN THE RANGE OF 35000 TO 45000?

NOT BETWEEN

----------

WHENEVER WE NEED TO EXCLUDE SOME RANGE OF VALUES WE USE NOT BETWEEN.

SYNTAX:

< >

COLUMN\_NAME/EXPRESSION NOT BETWEEN LOWER\_RANGE\_VALUE AND HIGHER\_RANGE\_VALUE;

SELECT \*

FROM EMPS

WHERE SAL NOT BETWEEN 35000 AND 45000;

33. WQTD DETAILS OF THE EMPS WHO WERE NOT JOINED IN THE YEAR 2020??

SELECT \*

FROM EMPS

WHERE DOJ NOT BETWEEN '2020-01-01' AND '2020-12-31';

34.WQTD DETAILS OF THE EMPS WHO ARE GETTING SOME COMMISSION???

IS

--

IT IS USED TO CHECK WHETHER THE COLUMN IS NULL OR NOT NULL

SYNTAX:

COLUMN\_NAME/EXPRESSION IS NULL/NOT NULL;

SELECT \*

FROM EMPS

WHERE COMM IS NOT NULL;

35.WQTD DETAILS OF THE EMPS WHO ARE ACTING AS CUSTOMERS FOR THEIR COMPANY????

SELECT \*

FROM EMPS

WHERE CID IS NOT NULL;

36.WQTD DETAILS OF THE EMPS THOSE WHO HAVE REPORTING MANAGER???

SELECT \*

FROM EMPS

WHERE MGR IS NOT NULL;

37. WQTD FNAME OF EMPS IF NAME IS STARTING WITH S?

A. SELECT FNAME FROM EMPS WHERE FNAME LIKE 'S%';

LIKE

----

38. WQTD DETAILS OF EMPS IF LNAME IS ENDING WITH I?

A. SELECT \* FROM EMPS WHERE LNAME LIKE '%I';

39. WQTD DETAILS OF EMPS IF NAME CONTAINS ATLEAST 2 A AND GETTING SALARY MORE THAN 25000?

A. SELECT \* FROM EMPS WHERE FNAME LIKE '%A%A%' AND SAL>25000;

40. WQTD FNAME, LANME, JOB IF JOB CONTAINS STRING MAN IN IT?

A. SELECT FNAME, LNAME, JOB FROM EMPS WHERE JOB LIKE '%MAN%';

41. WQTD DETAILS OF EMPLOYEE WHOSE FNAME STARTS WITH K OR A?

A. SELECT \* FROM EMPS WHERE FNAME LIKE 'K%' OR FNAME LIKE 'A%';

42. WQTD DETAILS OF EMPS IF EMP HIRED IN THE YEAR 2019?

A. SELECT \* FROM EMPS WHERE DOJ LIKE '2019%';

43. WQTD FNAME, LNAME, DOB IF EMPS BORN IN THE MONTH OF JAN, FEB OR MARCH?

A. SELECT FNAME, LNAME, DOB FROM EMPS WHERE DOB LIKE '%-01-%' OR DOB LIKE '%-02-%' OR DOB LIKE '%-03-%';

44. WQTD DETAILS OF EMPS THOSE WHO WERE JOINED IN THE DATE OF 12 OR 01?

A. SELECT \* FROM EMPS WHERE DOJ LIKE '%-12' OR DOJ LIKE '%-01';

45. WQTD FNAME FROM EMPS TABLE IF NAME IS NOT STARTING WITH S?

A. SELECT FNAME FROM EMPS WHERE FNAME NOT LIKE 'S%';

NOT LIKE

--------

IT IS USED FOR PATTERN MATCHING

>HERE, IT WILL EXCLUDE THE DETAILS BASED ON THE PATTERN.

SYNTAX

------

COLUMN\_NAME/EXPRESSION NOT LIKE 'PATTERN\_TO\_MATCH';

46. WQTD DETAILS OF THE EMPS THOSE WHO WERE NOT BORN IN THE YEAR 1995?

A. SELECT \* FROM EMPS WHERE DOB NOT LIKE '1995%';

47. WQTD DETAILS OF EMPS WHOSE FNAME IS NOT STARTING WITH S AND D?

A. SELECT \* FROM EMPS WHERE FNAME NOT LIKE 'S%' AND FNAME NOT LIKE 'D%';

48. WQTD FNAME LNAME IF LNAME

A. SELECT FNAME, LNAME FROM EMPS WHERE LNAME LIKE '\_\_\_';

49. WQTD DETAILS OF EMPS IF EMP FNAME LAST 3RD CHARACTER IS E AND LNAME LAST CHARACTER IS R AND WORKING AS WAITER OR MANAGER BUT NOT AS FEMALE?

A. SELECT \* FROM EMPS WHERE FNAME LIKE '%E\_\_' AND LNAME LIKE '%R' AND (JOB = 'MANAGER' OR JOB = 'WAITER') AND GENDER !='F';

50. WQTD DETAILS FROM PENTAGON TABLE IF NAME CONTAINS ATLEAST 1 % IN IT?

A. SELECT \* FROM PENTAGON WHERE NAME LIKE '%\%%';

'\' IT IS USED TO REMOVE SPECIAL BEHAVIOUR FROM SPECIAL CHARACTER.

51. WQTD NAME FROM PENTAGON TABLE IF NAME CONTAINS ATLEAST 2 % IN IT?

A. SELECT \* FROM PENTAGON WHERE NAME LIKE '%\%%\%%';

52. WQTD NAME FROM PENTAGON TABLE IF NAME 5TH CHARACTER IS \_ ?

A. SELECT NAME FROM PENTAGON WHERE NAME LIKE '\_\_\_\_\\_%';

53. WQTD DETAILS OF EMPS IF NAME IS STARTING WITH VOWELS?

A. SELECT \* FROM EMPS WHERE FNAME LIKE 'A%' OR FNAME LIKE 'E%' OR FNAME LIKE 'I%' OR FNAME LIKE 'O%' OR FNAME LIKE 'U%';

function

--------

set of instruction to perform specific tasks .

Function

|

+---------------+

| |

user Inbuilt func

defined fun

| |

1 procedure 1 aggregate function

2aggregate 2 character function

3 number function

4 date function

1 AGGREGATE FUNCTION/GROUP FUNCTION/MULTI ROW FUNCTION

-------------------------------------------------

TYPES OF AGGREGATE FUNCTION

-------------------------

1.MAX():

--------

IT IS USED TO OBTAIN MAXIMUM VALUE FROM THE GIVEN COLUMN WE USE MAX FUNCTION

SYNTAX : MAX(COLUMN\_NAME/EXPRESSON)

SELECT MAX(SAL) FROM EMPS;

----

2.MIN():

--------

IT IS USED TO OBTAIN MINIMUM VALUE FROM GIVEN COLUMN

SYNTAX: MIN(COLUMN\_NAME/EXPRESSION)

SELECT MIN(SAL) FROM EMPS;

---

3.AVG():

--------

IT IS USED TO OBTAIN AVERAGE VALUE FROM GIVEN COLUMN

SYNTAX: AVG(COLUMN\_NAME/EXPRESSION)

FOR EX:

SELECT AVG(SAL)

FROM EMPS;

---

4.SUM():

--------

IT IS USED TO OBTAIN SUM (TOTAL) VLUE FROM GIVEN COLUMN

SYNTAX: SUM(COLUMN\_NAME/EXPRESSION)

FOR EG:

SELECT SUM(SAL)

FROM EMPS;

---

5.COUNT():

----------

IT IS USED TO OBTAIN NUMBER OF VALUES PRESENT IN GIVEN COLUMN

SYNTAX: COUNT(\*/COLUMN\_NAME/EXPRESSION)

FOR EG:

SELECT COUNT(SAL)

FROM EMPS;

NOTE : ONLY FOR COUNT() WE CAN USE \* AS A ARGUMENT.

CHARACTERISTICS OF AGGREGATE FUNCTION

-------------------------------------

- IT TAKES N NUMBER OF INPUT BUT GENERATE SINGLE OUTPUT.

- IT EXECUTES GROUP BY GROUP.

- WE CAN'T USE NORMAL COLUMN ALONG WITH AGGREATE FUNCTION IN SELECT CLAUSE.

- WE CAN PASS ONLY ONE COLUMN AS A AGRUMENT INSIDE AGGREGATE FUNCTION.

- WE CAN'T NEST AGGREGATE FUNCTIONS.

- WE CAN'T USE AGGREGATE FUNCTION INSIDE WHERE CLAUSE.

- IT IGNORE NULL VALUES.

- WE CAN USE GROUP BY EXPESSION ALONG WITH AGGREGATE FUNCTION IN SELECT CLAUSE.

----------------------------------------------------------------------------------

54. WQTD MAX SALARY AND MIN SALARY FROM EMPS TABLES?

A. SELECT MAX(SAL),MIN(SAL) FROM EMPS;

55. WQTD TOTAL SALARY GIVEN TO ALL THE WAITERS?

A. SELECT SUM(SAL) FROM EMPS WHERE JOB= 'WAITER';

56. WQTD AVERAGE SALARY, TOTAL SALARY, MIN SAL, MAX SAL GIVEN TO THE EMPS WHOSE FNAME LAST 2ND CHARACTER IS S?

A. SELECT AVG(SAL), SUM(SAL), MIN(SAL), MAX(SAL) FROM EMPS WHERE FNAME LIKE '%S\_';

57. WQTD NUMBER OF EMPLOYESS WORKING AS DELIVERY OR MANAGER?

A. SELECT COUNT(JOB) FROM EMPS WHERE JOB IN ('DELIVERY', 'MANAGER');

58. WQTD NUMBER OF EMPS GETTING SALARY MORE THAN 50000 AND BORN DURING THE DATE 12 AND NOT GETTING COMMISSION?

A. SELECT COUNT(JOB) FROM EMPS WHERE SAL > 50000 AND DOJ LIKE '%-12' AND COMM IS NULL;

59. WQTD NUMBER OF UNIQUE JOB ROLES PRESENT IN EMPS TABLE?

A. SELECT COUNT(DISTINCT JOB) FROM EMPS;

60. WQTD NUMBER OF EMPLOYEES WORKING IN EACH JOB ROLE?

A.

GROUP BY

--------

IT IS USED TO CREATE A GROUP

CHARACTERRISTICS OF GROUP BY

----------------------------

IT EXECUTES ROW BY ROW.

WE CAN USE MULTIPLE COLUMN INSIDE GROUP BY CLAUSE, IT WILL CREATE A GROUP BASED ON COMBINATION OF THE COLUMNS.

IT EXECUTES ROW BY ROW BUT AFTER THE EXECUTION IT CREATE GROUP.

ANY CLAUSE WHICH EXECUTES AFTER GROUP BY CLAUSE WILL EXECUTE GROUP BY GROUP ONLY.

GROUP BY EXPRESSION

-------------------

THE COLUMNS WHICH WE ARE USING INSIDE GROUP BY CLAUSE WILL BE CONSIDERED AS GROUP BY EXPRESSION.

+--------------------------------+

|ID |NAME |SAL |JOB |

|1 |ROMEO |8000 |WAITER |

|2 |MAJNU |9000 |MANAGER |

|3 |JULIET |10000 |WAITER |

|4 |LAILA |7000 |MANAGER |

|5 |DEVDAS |4000 |SECURITY|

| | | | |

+--------------------------------+

ORDER OF EXECUTION

------------------

1. FROM

2. WHERE

3. GROUP BY

4. SELECT

61. WQTD TOTAL SALARY SPENT BY COMPANY IN EACH JOB ROLE IF JOB ARE DELIVERY, MANAGER OR WAITER?

A. SELECT JOB,SUM(SAL) FROM EMPS WHERE JOB IN ('DELIVERY', 'MANAGER', 'WAITER') GROUP BY JOB;

62. WQTD NUMBER OF EMPS WORKING IN EACH LOCATION AND GETTING SALARY MORE THAN 32000 AND LESS THAN 50000?

A.

63. WQTD

A.

HAVING

------

IT IS USED TO FILTER AGGREGATE FUNCTION

CHARACTERISTICS OF HAVING CLAUSE

--------------------------------

IT EXECUTES AFTER GROUP BY CLAUSE.

IT EXECUTES GROUP BY GROUP.

IT EVALUATE TRUE OR FALSE CONDITION.

WE CAN USE MULTIPLE CONDITION INSIDE HAVING CLAUSE.

WE CAN'T USE NORMAL COLUMNS INSIDE HAVING CLAUSE.

61. WQTD AVG SALARY AND TOTAL SALARY GIVEN TO EMPS IN EACH LOCATION IF AVG SALARY IS MORE THAN 40000?

A. SELECT AVG(SAL), SUM(SAL), LID FROM EMPS GROUP BY LID HAVING AVG(SAL)>40000;

65. WQTD TOTAL SALARY AND NUMBER OF EMPLOYEES WORKING IN EACH JOB ROLE IF MORE THAN 2 EMPS WORING IN EACH JOB ROLE?

A. SELECT JOB, SUM(SAL), COUNT(\*) FROM EMPS GROUP BY JOB HAVING COUNT(\*)>2;

66. WQTD MAXIMUM SALARY AND MINIMUM SALARY AND NUMBER OF EMPS WORKING IN EACH LOCATION IF THE LOCATION CONTAINS ATLEAST 2 EMPS WORKING IN IT AND EMP SALARY MUST BE MORE THAN 32000?

A. SELECT MAX(SAL), MIN(SAL), COUNT(\*), LID FROM EMPS WHERE SAL>32000 GROUP BY LID HAVING COUNT(\*)>=2;

67. WQTD NUMBER OF EMPS WHO ARE GETTING SAME SALARY/ REPEATED SALARY?

A. SELECT COUNT(\*) FROM EMPS GROUP BY SAL HAVING COUNT(\*)>1;

68. WQTD NUMBER OF EMPLOYEES HAVING SAME GENDER AND WORKING IN SAME JOB ROLE?

A. SELECT COUNT(\*), GENDER, JOB FROM EMPS GROUP BY GENDER, JOB HAVING COUNT(\*)>1;

69. WQTD DETAILS OF EMPS ACCORDING TO SAL MAXIMUM TO MINIMUM ORDER?

A. SELECT \* FROM EMPS ORDER BY SAL DESC;

ORDER BY

--------

IT IS USED TO ARRANGE THE RECORDS EITHER IN ASSCENDING ORDER OR DESCENDING ORDER.

SYNTAX:

SELECT COLUMN\_NAME/EXPRESSION

FROM TABLE\_NAME

ORDER BY COLUMN\_NAME ASC/DESC;

CHARACTERISTICS OF ORDER BY CLAUSE

----------------------------------

>IT IS A LAST EXECUTABLE CLAUSE IN A QUERY.

>IT EXECUTES AFTER SELECT CLAUSE.

>BY DEFAULT IT WILL CONSIDER ASC ORDER FOR A COLUMN.

>NORMALLY ALL THE RECORDS IN A TABLE ARRANGED IN ASC ORDER BASED ON PRIMARY KEY COLUMN.

>WE CAN USE ALIAS NAME INSIDE ORDER BY CLAUSE.

>WE CAN USE COLUMNS INSIDE ORDER BY CLAUSE, IT WILL GIVE THE PRIORITY FOR FIRST COLUMN ORDER, IF THE VALUES ARE SAME FOR FIRST COLUMN ORDER THAN IT GIVE THE PRIORITY FOR 2ND COLUMN ORDER.

ORDER OF EXECUTION

------------------

1. FROM

2. WHERE

3. GROUP BY

4. HAVING

5. SELECT

6. ORDER BY

70. WQTD FNAME, LNAME, JOB IF EMPLOYEE IS WORKING AS SECURITY OR MANAGER OR CLEANER AND ARRANGE THE RECORDS ACCORDING TO ALPHABETICAL ORDER OF FNAME?

A. SELECT FNAME, LNAME, JOB FROM EMPS WHERE JOB IN('SECURITY', 'MANAGER', 'CLEANER') ORDER BY FNAME ASC;

71. WQTD NUMBER OF EMPS WHO ARE HAVING SAME GENDER AND WORKING IN SAME JOB ROLE AND EMPS ARE GETTING SAL MORE THAN 30000 AND ARRANGE THE JOB IN ALPHA ORDER?

A. SELECT COUNT(\*), JOB, GENDER FROM EMPS WHERE SAL>30000 GROUP BY GENDER, JOB HAVING COUNT(\*)>1 ORDER BY JOB ASC;

72. WQTD DETAILS OF FIRST 3 RECORDS FROM EMPS TABLE?

A. SELECT \* FROM EMPS LIMIT 3;

LIMIT

-----

IT IS USED TO DISPLAY SOME SPECIFIC NUMBER OF RECORDS FROM RESULTANT TABLE.

SYNTAX:

-------

SELECT COLUMN\_NAME/EXPRESSION

FROM TABLE\_NAME

LIMIT VALUE;

73. WQTD DETAILS OF FIRST RECORD FROM EMPS TABLE?

A. SELECT \* FROM EMPS LIMIT 1;

74. WQTD DETAILS OF 2ND RECORD FROM EMPS TABLE?

A. SELECT \* FROM EMPS LIMIT 1,1;

OFFSET

------

IT IS USED TO SKIP OR IGNORE SOME SPECIFIC NUMBER OF RECORDS FROM RESULTANT TABLE

SYNTAX:

-------

SELECT COLUMN\_NAME/EXPRESSION

FROM TABLE\_NAME

LIMIT VALUE OFFSET VALUE;

75. WQTD DETAILS OF 5TH AND 6TH RECORDS FROM EMPS?

A. SELECT \* FROM EMPS LIMIT 2 OFFSET 4;

76. WQTD DETAILS OF 4TH AND 6TH RECORDS?

A.

77. WQTD DETAILS OF TOP 5 MAX SALARY HOLDER?

A. SELECT \* FROM EMPS ORDER BY SAL DESC LIMIT 5;

78. WQTD LAST 3 RECORDS FROM EMPS TABLE?

A. SELECT \* FROM EMPS ORDER BY EID DESC LIMIT 3;

79. WQTD 3RD MAX SAL FROM EMPS TABLE?

A. SELECT DISTINCT SAL FROM EMPS ORDER BY SAL DESC LIMIT 1 OFFSET 2;

80. WQTD 4TH MINIMUM SALARY FROM EMPS TABLE?

A. SELECT DISTINCT SAL FROM EMPS ORDER BY SAL ASC LIMIT 1 OFFSET 3;

CHARACTER FUNCTION

------------------

CHARACTERISTICS OF CHARACTER FUNCTION

-------------------------------------

>IT TAKES N NUMBER OF INPUTS AND GENERATE N NUMBER OF OUTPUTS.

>IT EXECUTE ROW BY ROW.

>WE CAN NEST CHARACTER FUNCTIONS.

>WE CAN USE CHARACTER FUNCTION INSIDE WHERE CLAUSE.

TYPES OF CHARACTER FUNCTIONS

----------------------------

1. LOWER()

----------

IT IS USED TO CONVERT GIVEN STRING VALUE INTO LOWER CASE.

EX:

SELECT LOWER('INDIA'); \\OP: india

2. UPPER()

----------

IT IS USED TO CONVERT GIVEN STRING VALUE INTO UPPER CASE.

EX:

SELECT UPPER('india'); \\OP: INDIA

3. LENGTH()

-----------

IT IS USED TO OBTAIN TOTAL NUMBER OF CHARACTERS PRESENT IN GIVEN STRING VALUE.

EX:

SELECT LENGTH('PENTAGON'); \\OP: 8

SELECT LENGTH('PENTAGON SPACE') \\OP: 14

81. WQTD SECOND LONGEST CITY AS WELL AS THEIR RESPECTIVE LENGTH FROM LOCATION TABLE IF THERE IS MORE THAN ONE SECOND LONGEST CITY CHOOSE THE ONE WHICH COMES FIRST WHEN WE ORDER ALPHABETICALLY?

A. SELECT CITY, LENGTH(CITY) FROM LOCATIONS GROUP BY CITY ORDER BY LENGTH(CITY) DESC, CITY ASC LIMIT 1 OFFSET 1;

4. REVERSE()

------------

IT IS USED TO DISPLAY GIVEN STRING VALUE IN REVERSE FORMAT.

EX:

SELECT REVERSE('INDIA'); \\OP: AIDNI

5. CONCAT()

-----------

IT IS USED TO COMBINE OR ADD 2 OR MORE STRING VALUE.

SYNTAX: CONCAT('STR\_1', 'STR\_2'...,'STR\_N');

SAMPLE OP:

MR/MISS SHAKILA YOUR SALARY IS 500 RS.

82. WQTD FNAME, LNAME AND JOB TOGETHER IN BELOW FORMAT:

FNAME: SHAHRUKH

LNAME: KHAN

JOB: ACTOR

OP: SHAHRUKKH KHAN (ACTOR)

A. SELECT CONCAT(FNAME, ' ', LNAME , ' ', '(',JOB,')') FROM EMPS;

6. SUBSTR()

-----------

IT IS USED TO EXTRACT SOME PART OF THE STRING FROM ORIGINAL STRING.

SYNTAX: SUBSTR('ORIGINAL STRING', POSITION, [LENGTH]); \\LENGTH IS OPTIONAL.

CASE 1:

-------

B E N G A L U R U

1 2 3 4 5 6 7 8 9

SELECT SUBSTR('BENGALURU', 5, 3); OP: ALU

SELECT SUBSTR('BENGALURU', 6); OP: LURU

CASE 2:

-------

B E N G A L U R U

-9 -8 -7 -6 -5 -4 -3 -2 -1

SELECT SUBSTR('BENGALURU', -3, 2); OP: UR

SELECT SUBSTR('BENGALURU', -8); OP: ENGALURU

83. WQTD DETAILS OF THE EMPS IF EMPLOYEE FNAME IS STARTING WITH 'A'?

A. SELECT \* FROM EMPS WHERE SUBSTR(FNAME,1,1)='A';

84. WQTD FNAME, LNAME IF EMPLOYEE LNAME IS ENDING WITH 'I'?

A. SELECT FNAME, LNAME FROM EMPS WHERE SUBSTR(LNAME, -1,1)='I';

85. WQTD DETAILS OF THE EMPLOYEE IF THERE JOB IS NOT ENDING WITH TER?

A. SELECT \* FROM EMPS WHERE SUBSTR(JOB,-3,3) != 'TER';

86. WQTD DETAILS OF EMPOLYEE IF FNAME STARTING WITH 'S' OR 'A'?

A. SELECT \* FROM EMPS WHERE SUBSTR(FNAME, 1, 1) IN ('A','S');

87. WQTD FNAME, LNAME, JOB IF JOB STARTING WITH SEC OR WAIT?

A. SELECT FNAME, LNAME, JOB FROM EMPS WHERE SUBSTR(JOB,1,3)='SEC' OR SUBSTR(JOB,1,4)='WAIT';

88. WQTD DETAILS OF EMPLOYEE WHOSE FNAME IS NOT STARTING WITH VOWELS?

A. SELECT \* FROM EMPS WHERE SUBSTR(FNAME, 1, 1) NOT IN ('A','E', 'I', 'O', 'U');

89. WQTD FNAME, LNAME, SALARY AND DOB IF EMPLOYEE BORN DURING 1995?

A. SELECT FNAME, LNAME, DOB FROM EMPS WHERE SUBSTR(DOB,1,4)='1995';

90. WQTD DETAILS OF THE EMPS IF EMPLOYEE JOINED IN THE MONTH OF APRIL, MAY, JUNE OR JULY?

A. SELECT \* FROM EMPS WHERE SUBSTR(DOJ, 6, 2) IN (04, 05, 06, 07);

91. WQTD FIRST HALF OF FNAME FROM EMPS TABLE?

A. SELECT SUBSTR(FNAME, 1, LENGTH(FNAME)/2) FROM EMPS;

92. WQTD SECOND HALF OF FNAME FROM EMPS TABLE?

A. SELECT SUBSTR(FNAME, LENGTH(FNAME)/2+1) FROM EMPS;

93. WQTD FIRST HALF OF JOB IN LOWERCASE AND SECOND HALF OF JOB IN REVERSE FORMAT?

A. SELECT CONCAT (LOWER(SUBSTR(JOB, 1,LENGTH(JOB)/2)), REVERSE(SUBSTR(JOB, LENGTH(JOB)/2+1))) AS JOB FROM EMPS;

94. WQTD FNAME AND PASSWORD, PASSWORD MUST CONTAIN BELOW CONDITIONS:

i. FIRST 3 CHARACTERS OF FNAME

ii. LENGTH OF STATUS

iii. LAST 3 CHARACTERS OF THERE JOB ROLE.

I/P:

FNAME: MANJA

STATUS: BUSY

JOB ROLE: WAITER

O/P:

FNAME: MANJA

PASSWORD: MAN4TER

A. SELECT FNAME, CONCAT (SUBSTR(FNAME,1,3),LENGTH(STATUS),(SUBSTR(JOB,-3,3))) AS PASSWORD FROM EMPS;

95. WQTD FNAME AND PASSWORD FOR KIRAN, PASSWORD MUST CONTAIN

i. 2ND HALF OF FNAME

ii. LENGTH OF JOB ROLE

iii. REVERSE LAST 4 CHARACTER OF HIS STATUS.

A. SELECT FNAME, CONCAT (SUBSTR(FNAME,LENGTH(FNAME)/2+1),LENGTH(JOB),REVERSE(SUBSTR(STATUS,-4,4))) AS PASSWORD FROM EMPS WHERE FNAME='KIRAN';

REPLACE FUNCTION

----------------

IT IS USED TO REPLACE A SUBSTRING FROM NEW STRING IN ORIGINAL STRING.

SYNTAX:

-------

REPLACE('ORIGINAL STRING','SUB STRING','NEW STRING');

SEECT REPLACE('PENTAGON', 'PENT', 'HEX');

SELECT REPLACE('PENTAGON','N','S');

SELECT REPLACE('PENTAGON','R','S');

SELECT REPLACE('PENTAGON','N',' ');

SELECT REPLACE('PENTAGON', 'n', 'R');

96. WQT REPLACE A AND I WITH [A] AND [I] IN FNAME OF THE EMPLOYEES?

FNAME: KIRAN

O/P: K[I]R[A]N

A. SELECT REPLACE(REPLACE(FNAME,'A','[A]'), 'I','[I]') FROM EMPS;

97. WQTD COUNT OF A CHARACTER A IN MALAYALAM?

A. SELECT LENGTH('MALAYALAM')-LENGTH(REPLACE('MALAYALAM','A',''));

98. WQTD DETAILS OF EMPLOYEES IF FNAME CONTAINS EXACTLY ONE A?

A. SELECT \* FROM EMPS WHERE LENGTH(FNAME)-LENGTH(REPLACE(FNAME,'A',''))=1;

NUMBER FUNCTIONS

----------------

CHARACTERISTICS OF NUMBER FUNCTION

----------------------------------

>IT TAKES N NUMBER OF INPUTS AND GENERATE N NUMBER OF OUTPUTS

>IT EXECUTES ROW BYY ROW

>WE CAN NEST NUMBER FUNCTIONS

>WE CAN USE NUMBER FUNCTION INSIDE WHERE CLAUSE

TYPES OF NUMBER FUNCTIONS

-------------------------

1. ABS()

---------

IT IS USED TO CONVERT NEGATIVE VALUE TO POSITIVE VALUE.

EX:

SELECT ABS(-45); OP:45

SELECT ABS(45); OP:45

2. MOD()

--------

IT IS USED TO OBATIN REMAINDER VALUE.

SYNTAX: MOD(M,D);

SELECT MOD(8,2); OP:0

SELECT MOD(8,3); OP:3

99. WQTD DETAILS OF EMPS WHO ARE HAVING EVEN EID?

A. SELECT \* FROM EMPS WHERE MOD(EID,2)=0;

3. ROUND()

----------

IT IS USED TO ROUND OF A NUMMBER UPTO SPECIFIED NUMBER OF DECIMAL PLACES.

SYYNTAX: ROUND(NUMBER, DECIMAL\_PLACE\_VALUE);

SELECT ROUND(123.4); OP:123

SELECT ROUND(123.6); OP:124

SELECT ROUND(123.456, 2) OP:123.46

SELECT ROUND(123.45645, 3) OP:123.456

100. WQTD AVG SALARY OBTAINED IN EACH JOB AND ROUND OFF THE NUMBER UPTO 2ND DECIMAL PLACE?

A. SELECT ROUND(AVG(SAL),2) AS 'AVERAGE SALARY', JOB FROM EMPS GROUP BY JOB;

4. CEIL()

---------

IT WILL OBTAIN NEXT INTEGER VALUE FROM THE GIVEN DECIMAL VALUE(IF IT IS POSITIVE NUMBER).

IT WILL OBTAIN CURRENT INTEGER VALUE FROM GIVEN DECIMAL VALUE(IF IT IS NEGATIVE NUMBER).

SEELCT CEIL(4.3); OP:5

SELECT CEIL(-4.9); OP:-4

5.FLOOR()

---------

IT WILL OBTAIN CURRENT INTEGER VALUE FROM GIVEN DECIMAL VALUE(IF IT IS POSITIVE NUMBER).

IT WILL OBTAIN NEXT INTEGER VALUE FROM GIVEN DECIMAL VALUE(IF IT IS NEGATIVE NUMBER).

SELECT FLOOR(5.9); OP:5

SELECT FLOOR(-5.9); OP:-6

6. TRUNCATE()

-------------

IT IS USED TO CUTOFF A NUMBER UPTO SPECIFIED NUMBER OF DECIMAL PLACES WITHOUT ROUNDING IT.

SYNTAX: TRUNCATE(NUMBER,DECIMAL\_PLACE\_VALUE);

SELECT TRUNCATE(143.4567,2); OP:143.45

SELECT TRUNCATE(143.4567,3); OP:143.456

7. POW()

--------

IT IS USED TO OBTAIN POWER VALUE OF A NUMBER.

EX: SELECT POW(8,2); OP:64

8. SQRT()

--------

IT IS USED TO OBATIN SQUARE ROOT VALUE OF NON NEGATIVE NUMBER.

EX: SELECT SQRT(64); OP:8

SELECT SQRT(-64); OP:NULL

DATE FUNCTIONS

--------------

CHARACTERISTICS OF DATE FUNCTIONS

---------------------------------

>IT TAKES N NUMBER OF INPUT AND GENERATE N NUMBER OF OUTPUTS.

>IT EXECUTES ROW BY ROW.

>WE CAN NEST DATE FUNCTIONS.

>WE CAN USE DATE FUNCTIONS INSIDE WHERE CLAUSE.

TYPES OF DATE FUNCTIONS

------------------------

1. CURDATE()

------------

IT IS USED TO OBTAIN CURRENT DATE FROM THE SYSTEM.

SELECT CURDATE(); OP:2025-07-10

2. SYSDATE()/NOW()

------------------

IT IS USED TO OBTAIN CURRENT DATE AND TIME FROM THE SYSTEM.

SELECT SYSDATE(); OP:2025-07-10 16:08:59

SELECT NOW(); OP:2025-07-10 16:08:59

3. YEAR()

---------

IT IS USED TO EXTRACT YEAR FROM GIVEN DATE EXPRESSION.

SELECT YEAR('2020-10-11'); OP:2020

SELECT YEAR(DOB) FROM EMPS; OP

4. MONTH()

----------

IT IS USED TO EXTRACT MONTH FROM GIVEN DATE EXPRESSION.

SELECT('2021-12-11'); OP:12

5. DAY()

--------

IT IS USED TO EXTRACT DATE VALUE FROM GIVEN DATE EXPRESSION.

SELECT('2012-12-30'); OP:30

101. WQTD DETAILS OF EMPLOYEES BORN BEFORE THE YEAR 1994?

A. SELECT \* FROM EMPS WHERE YEAR(DOB)<1994;

102. WQTD DETAILS OF EMPLOYEES IF EMP HIRED IN THE MONTH OF JAN, FEB OR APRIL?

A. SELECT \* FROM EMPS WHERE MONTH(DOJ) IN (01,02,04);

103. WQTD DETAILS OF EMPLOYEES IF EMP HIRED IN LEAP YEAR?

A. SELECT \* FROM EMPS WHERE MOD(YEAR(DOJ),4)=0;

104. WQTD FNAME, CURRENT EXPERIENCE OF EMPLOYEES IN TERMS OF YEARS?

A. SELECT FNAME, YEAR((CURDATE()))-YEAR(DOJ) AS 'EMPLOYEE EXPERIENCE'FROM EMPS;

105. WQTD LAST HIRED EMPLOYEE DOJ?

A. SELECT DISTINCT DOJ FROM EMPS ORDER BY DOJ DESC LIMIT 1;

6. DATEDIFF()

-------------

IT IS USED TO OBTAIN DAY DIFFERENCE BETWEEN TWO DATE VALUES.

SELECT DATEDIFF(CURDATE(),'2025-07-09'); OP:2

7. DATE\_ADD()

-------------

IT IS USED TO ADD SOME TIME INTERVAL FOR A GIVEN DATE.

SYNTAX:

DATE\_ADD('DATE VALUE', INTERVAL VALUE UNIT);

INTERVAL: IT IS A KEYWORD USED TO ADD/SUBTRACT SOME TIME INTERVAL.

VALUE: THE AMOUNT OF TIME WHICH WE ARE ADDING.

UNIT: UNIT OF TIME INTERVAL(EX: YEAR,MONTH,DAY ETC...).

106. WQT ADD 6 YEAR FOR BELOW DATE:

'2000-10-12'

A. SELECT DATE\_ADD('2000-10-12',INTERVAL 6 YEAR);

107. WQT TO ADD 4 YEAR 7 MONTHS FOR BELOW DATE:

'2020-03-07'

A. SELECT DATE\_ADD(DATE\_ADD('2020-03-07',INTERVAL 4 YEAR),INTERVAL 7 MONTH); OR SELECT DATE\_ADD('2020-03-07',INTERVAL 55 MONTH);

8. DATE\_SUB()

-------------

IT IS USED TO SUBTRACT SOME TIME INTERVAL FROM GIVEN DATE VALUE.

SYNTAX: DATE\_SUB('DATE VALUE',INTERVAL VALUE UNIT);

108. WQT SUBTRACT 5 DAYS FROM CURRENT DATE?

A. SELECT DATE\_SUB(CURDATE(),INTERVAL 5 DAY);

109. WQT SUBTRACT 2 YEAR 4 MONTH FROM FIRST HIRED EMPLOYEE DOJ?

A. SELECT DATE\_SUB(MIN(DOJ), INTERVAL 28 MONTH) FROM EMPS; OR SELECT DATE\_SUB(DOJ, INTERVAL 28 MONTH) FROM EMPS ORDER BY DOJ ASC LIMIT 1;

9. DATE\_FORMAT();

-----------------

IT IS USED TO EXTRACT INDIVIDUAL CHARACTERS FROM GIVEN DATE-TIME EXPRESSION.

SYNTAX: DATE\_FORMAT('DATE VALUE','DATE\_FORMAT\_PATTERN');

YYYY: '%Y' SELECT DATE\_FORMAT(NOW(),'%Y'); OP:2025

YYYY: '%y' SELECT DATE\_FORMAT(NOW(),'%y'); OP:25

MONTH: '%M' SELECT DATE\_FORMAT(NOW(),'%M'); OP:JULY

MM: '%m' SELECT DATE\_FORMAT(NOW(),'%m'); OP:07

MON: '%b' SELECT DATE\_FORMAT(NOW(),'%b'); OP:JUL

DAY: '%W' SELECT DATE\_FORMAT(NOW(),'%W'); OP:FRIDAY

DAY: '%a' SELECT DATE\_FORMAT(NOW(),'%a'); OP:FRI

DD: '%d' SELECT DATE\_FORMAT(NOW(),'%d'); OP:11

HH24: '%H' SELECT DATE\_FORMAT(NOW(),'%H'); OP:16

HH12: '%h' SELECT DATE\_FORMAT(NOW(),'%h'); OP:04

MINUTE:'%i' SELECT DATE\_FORMAT(NOW(),'%i'); OP:27

SECOND: '%s' SELECT DATE\_FORMAT(NOW(),'%s'); OP:05

TIME: '%T' SELECT DATE\_FORMAT(NOW(),'%T'); OP:16:28:36

AM/PM: '%p' SELECT DATE\_FORMAT(NOW(),'%p'); OP:PM

TIME AM/PM:'%r' SELECT DATE\_FORMAT(NOW(),'%r'); OP:04:30:08 PM

110. WQT EXTRACT INDIVIDUAL CHARACTERS FROM CURRENT DATE AND TIME IN BELOW FORMAT

25-JUL-11 4 PM FRIDAY

A. SELECT DATE\_FORMAT(NOW(),'%y-%b-%d %h %p %W');

111. WQTD DETAILS OF EMPS IF EMPS WHERE JOINED IN DAYS OF FRIDAY, SATURDAY OR SUNDAY?

A. SELECT \* FROM EMPS WHERE DATE\_FORMAT(DOJ,'%W') IN ('FRIDAY', 'SATURDAY', 'SUNDAY');

112. WQTD NUMBER OF EMPS HIRED IN EACH YEAR AND DISPLAY THE NUMBERS MAX TO MIN ORDER BASED ON THEIR COUNT?

A. SELECT COUNT(\*), YEAR(DOJ) FROM EMPS GROUP BY YEAR(DOJ) ORDER BY COUNT(\*) DESC;

SUBQUERY

--------

IT IS A QUERY INSIDE ANOTHER QUERY.

WORKING PRINCIPLE OF SUB QUERY

------------------------------

>ALWAYS INNER QUREY EXECUTES FIRST AND WRITTEN SOME OUTPUT.

>THE OUTPUT OF INNER QUERY WILL BE GIVEN AS INPUT FOR THE OUTER QUERY.

>BY TAKING INPUT FROM THE INNER QUERY OUTER QUERY EXECUTES COMPLETLY AND RETURNS FINAL OUTPUT.

>OUTER QUERY DEPENDS ON INNER QUERY.

FINAL OP<-----OUTER QUERY----IP-

| |

------------------ |

| |

\/ |

OP<-------INNER QUERY <--

CASE 1: WHENEVER WE COME ACROSS SOME UNKNOWN VALUES WE USE SUBQUERY

EMP: ID NAME SAL

1 DHONI 800

2 KOHLI 900

3 ROHIT 600

4 HARDIK 500

Q. WQTD DETAILS OF EMPS IF EMPS ARE GETTING SALARY MORE THAN ROHIT?

A. SELECT \* FROM EMP WHERE SAL> (SELECT SAL FROM EMP WHERE NAME='ROHIT');

RULES

-----

1. THE COLUMN\_NAME SELECTED IN INNER QUERY AND THE COLUMN\_NAME WRITTEN INSIDE OUTER QUERY SHOULD BE OF SAME DATATYPE.

EX:

WHERE SAL>(SELECT SAL) VALID

INT INT

WHERE SAL>(SELECT COMM) VALID

INT INT

WHERE SAL>(SELECT NAME) INVALID

INT VARCHAR(10)

2. WE CAN SELECT ONLY ONE COLUMN INSIDE INNER QUERY.

EX:

WHERE SAL>(SELECT SAL,COMM) INVALID

WHERE SAL>(SELECT SAL) VALID

113. WQTD DETAILS OF EMPS WHO ARE GETTING SALARY LESS THAN PRIYA?

A. SELECT \* FROM EMPS WHERE SAL < (SELECT SAL FROM EMPS WHERE FNAME='PRIYA');

114. WQTD FNAME,LNAME,JOB AND SAL IF THE EMPS ARE WORKING SAME AS MURALI'S JOB ROLE?

A. SELECT FNAME,LNAME,JOB, SAL FROM EMPS WHERE FNAME !='MURALI' AND JOB=(SELECT JOB FROM EMPS WHERE FNAME='MURALI');

115. WQTD DETAILS OF EMPLOYEES WHO ARE HIRED AFTER 2 YEARS OF THE FIRST HIRED EMPLOYEE DOJ?

A. SELECT \* FROM EMPS WHERE DOJ > (SELECT DATE\_ADD(MIN(DOJ), INTERVAL 2 YEAR) FROM EMPS );

116. WQTD DETAILS OF EMPS IF EMPLOYEE FNAME 2ND CHARACTER IS A AND EMPLOYEE LNAME LAST 2ND CHARACTER IS A AND GEETING SALARY LESS THAN JAHNAVI?

A. SELECT \* FROM EMPS WHERE SUBSTR(FNAME,2,1)='A' AND SUBSTR(LNAME,-2,1)='A' AND SAL<(SELECT SAL FROM EMPS WHERE FNAME='JAHNAVI');

OR

SELECT \* FROM EMPS WHERE FNAME LIKE '\_A%' AND LNAME LIKE '%A\_' AND SAL <(SELECT SAL FROM EMPS WHERE FNAME='JAHNAVI');

117. WQTD FNAME, LNAME TOGETHER AS FULL NAME ALONG WITH SALARY, JOB, LID IF EMPS IS WORKING IN LOCATION SAME AS SURESH'S LOCATION AND GETTING SALARY MORE THAN PRIYA?

A. SELECT CONCAT(FNAME,' ', LNAME) AS 'FULL NAME', SAL, JOB, LID FROM EMPS WHERE LID =(SELECT LID FROM EMPS WHERE FNAME='SURESH') AND SAL >(SELECT SAL FROM EMPS WHERE FNAME='PRIYA');

118. WQTD DETAILS OF THE EMPLOYEES WHO ARE GETTING 2ND MINIMUM SALARY?

A. SELECT \* FROM EMPS WHERE SAL = (SELECT DISTINCT SAL FROM EMPS ORDER BY SAL ASC LIMIT 1 OFFSET 1 );

119. WQTD DETAILS OF THE EMPLOYEES WHO ARE ELDER THAN DIVYA?

A. SELECT \* FROM EMPS WHERE DOB <(SELECT DOB FROM EMPS WHERE FNAME='DIVYA');

CASE 2: WHENEVER DATA PRESENT IN ONE TABLE AND CONDITION GIVEN FROM ANOTHER TABLE WE USE SUBQUERY.

EMP: +-------------------+ +----+-------+

|ID |NAME |DNO| DEPT: |DNO | DNAME|

|1 |A |10 | +----+-------+

|10 | X |

|2 |B |20 | |20 | Y |

|3 |C |10 | |30 | Z |

|4 |D |30 | +----+-------+

+-------------------+

Q. WQTD DNAME OF B

A. SELECT DNAME FROM DEPT WHERE DNO= (SELECT DNO FROM EMP WHERE NAME='B')

120. WQTD CITY NAME OF EMPLOYEE KIRAN?

A. SELECT CITY FROM LOCATIONS WHERE LID=(SELECT LID FROM EMPS WHERE FNAME='KIRAN');

121. WQTD DETAILS OF EMPLOYEE WHO ARE LIVING IN THE STATE KARNATAKA?

A. SELECT \* FROM EMPS WHERE LID=(SELECT LID FROM LOCATIONS WHERE STATE='KARNATAKA');

122. WQTD CITY AND STATES OF THE CUSTOMER VIRAT KOHLI?

A. SELECT CITY, STATE FROM LOCATIONS WHERE LID=(SELECT LID FROM CUSTOMERS WHERE NAME='VIRAT KOHLI');

123. WQTD MENU ITEMS DETAILS ORDERED FROM SPICE HUB RESTAURANT?

A. SELECT \* FROM MENU\_ITEMS WHERE RESTAURANT\_ID=(SELECT RESTAURANT\_ID FROM RESTAURANTS WHERE NAME='SPICE HUB');

124. WQTD DETAILS OF EMPS WHO ARE WORING AS SECURITY OR MANAGER IN MUMBAI CITY?

A. SELECT \* FROM EMPS WHERE JOB IN ('SECURITY', 'MANAGER') AND LID = (SELECT LID FROM LOCATIONS WHERE CITY='MUMBAI');

125. WQTD DETAILS OF THE CUSTOMERS WHO ARE LIVING IN CHENNAI OR JAIPUR CITY?

A. SELECT \* FROM CUSTOMERS WHERE LID IN (SELECT LID FROM LOCATIONS WHERE CITY IN ('JAIPUR', 'CHENNAI'));

ERROR 1242 (21000): Subquery returns more than 1 row

----------------------------------------------------

TYPES OF SUBQUERY

-----------------

1. SINGLE ROW SUBQUERY

2. MULTI ROW SUBQUERY

3. CO-RELATED SUBQUERY

1. SINGLE ROW SUBQUERY

----------------------

IF INNER QUERY RETURNS SINGLE VALUE WE CAN CONSIDER THAT QUERY AS SINGLE ROW SUBQUERY.

>HERE WE CAN USE BOTH NORMAL OPERATORS(=,!=,>,<) AND SPECIAL OPERATORS(IN, NOT IN, ALL, ANY)

2. MULTI ROW SUBQUERY

---------------------

IF INNER QUERY RETURNS MORE THAN ONE VALUE WE CAN CONSIDER THAT QUERY AS MULTI ROW SUBQUERY.

>HERE WE CAN USE ONLY SPECIAL OPERATORS(IN, NOT IN, ALL, ANY).

126. WQTD DETAILS OF THE CUSTOMERS WHO ORDERED SOME PRODUCTS?

A. SELECT \* FROM CUSTOMERS WHERE ORDER\_ID IN (SELECT ORDER\_ID FROM ORDERS);

OR SELECT \* WHERE ORDER\_ID IS NOT NULL;

\*127. WQTD DETAILS OF EMPS WHO DELIVERED THE PRODUCT TO CUSTOMER WHO BELONGED TO KOLKATA CITY?

A. SELECT \* FROM EMPS WHERE EID IN (SELECT EID FROM ORDERS WHERE STATUS='DELIVERED' AND ORDER\_ID IN (SELECT ORDER\_ID FROM CUSTOMERS WHERE LID IN (SELECT LID FROM LOCATIONS WHERE CITY='KOLKATA')));

128. WQTD RESTAURANT NAME WHERE ALL ORDERS HAVE BEEN DELIVERED?

A. SELECT NAME FROM RESTAURANTS WHERE RESTAURANT\_ID IN (SELECT RESTAURANT\_ID FROM ORDERS WHERE STATUS='DELIVERED');

129. WQTD NAMES OF ALL THE CUSTOMERS WHOSE ORDER FAILED IN PAYMENT?

A. SELECT NAME FROM CUSTOMERS WHERE ORDER\_ID IN (SELECT ORDER\_ID FROM PAYMENTS WHERE STATUS='FAILED');

130. WQTD RESTAURANTS NAME THAT HAVE NEVER RECEIVED A REVIEWS?

A. SELECT NAME FROM RESTAURANTS WHERE RESTAURANT\_ID NOT IN (SELECT RESTAURANT\_ID FROM REVIEWS);

EMPLOYEE AND MANAGER RELATIONSHIP

---------------------------------

CASE 1: TO FIND MANAGER DETAILS

EMP: +------+--------+------+

|EID | NAME | MGR |

+------+--------+------+

| 1 | A | 2 |

| 2 | B | 3 |

| 3 | C | 4 |

| 4 | D | NULL |

+------+--------+------+

Q. WQTD MANAGER NAME OF C?

A. SELECT NAME FROM EMPS WHERE EID=(SELECT MGR FROM EMPS WHERE NAME='C')

131. WQTD MANAGER DETAILS OF THE EMPLOYEE SURESH?

A. SELECT \* FROM EMPS WHERE EID=(SELECT MGR FROM EMPS WHERE FNAME='SURESH');

132. WQTD CITY NAME OF DIVYA'S MANAGER?

A. SELECT CITY FROM LOCATIONS WHERE LID IN(SELECT LID FROM EMPS WHERE EID IN(SELECT MGR FROM EMPS WHERE FNAME='DIVYA'));

133. WQTD DETAILS OF AMAN'S MANAGER'S MANAGER?

A. SELECT \* FROM EMPS WHERE EID =(SELECT MGR FROM EMPS WHERE EID=(SELECT MGR FROM EMPS WHERE FNAME='AMAN'));

CASE 2: TO FIND EMPLOYEE DETAILS

--------------------------------

Q. WQTD EMPLOYEE NAME IF EMPLOYEE IS REPORTING TO D?

A. SELECT NAME FROM EMPS WHERE MGR=(SELECT EID FROM EMPS WHERE NAME='D')

134. WQTD DETAILS OF THE EMPLOYEES WHO ARE REPORTING TO JAHNAVI?

A. SELECT \* FROM EMPS WHERE MGR=(SELECT EID FROM EMPS WHERE FNAME='JAHNAVI');

135. WQTD DETAILS OF THE EMPLOYEES WHO ARE REPORTING TO ARJUN'S MANAGER?

A. SELECT \* FROM EMPS WHERE MGR =(SELECT EID FROM EMPS WHERE EID=(SELECT MGR FROM EMPS WHERE FNAME='ARJUN'));

136. WQTD LOCATION DETAILS OF EMPLOYEE WHO ARE REPORTING TO FAIZAN'S MANAGER'S MANAGER?

A. SELECT \* FROM LOCATIONS WHERE LID IN(SELECT LID FROM EMPS WHERE MGR IN (SELECT EID FROM EMPS WHERE EID IN (SELECT MGR FROM EMPS WHERE EID IN (SELECT MGR FROM EMPS WHERE FNAME='FAIZAN'))));

137. WQTD DETAILS OF EMPS WHO HAS WORKING EXPERIENCE MORE THAN MURALI BUT LESS THAN KIRAN???

A.SELECT \* FROM EMPS WHERE DATEDIFF(CURDATE(),DOJ)>(SELECT DATEDIFF(CURDATE(),DOJ) FROM EMPS WHERE FNAME='MURALI') AND DATEDIFF(CURDATE(),DOJ)<(SELECT DATEDIFF(CURDATE(),DOJ) FROM EMPS WHERE FNAME='KIRAN');

SELECT \* FROM EMPS WHERE DOJ<(SELECT DOJ FROM EMPS WHERE FNAME='MURALI') AND DOJ>(SELECT DOJ FROM EMPS WHERE FNAME='KIRAN');

138. WQTD DETAILS OF THE EMPS WHO ARE GETTING SALARY LESS THAN ALL THE WAITER??

A. SELECT \* FROM EMPS WHERE SAL<ALL(SELECT SAL FROM EMPS WHERE JOB='WAITER');

ALL

---

IT IS A MULTIVALUE OPERATOR WHICH TAKES MULTIPLE VALUES AT THE RHS AND SINGLE VALUE AT THE LHS ALONG WITH RELATIONAL OPERATORS.

SYNTAX:

LHS RHS

COLUMN\_NAME/EXPRESSION >/</>=/<= ALL(V1,V2,.......VN);

>IT WORKS ON AND CONDITION.

A:1000 T

B:2000 F

SAL<ALL(2000,3000,4000,5000)

1000<ALL(

2000<ALL(

SELECT \*

FROM EMPS

WHERE SAL<(SELECT MIN(SAL)

FROM EMPS

WHERE JOB='WAITER');

139.WQTD DETAILS OF THE EMPS IF EMPLOYEES ARE GETTING SALARY MORE THAN ANY ONE OF THE WAITER???

ANY:

----

IT IS A MULTI VALUE OPERATOR WHICH TAKES MULTIPLE VALUES AT THE RHS AND SINGLE VALUE AT THE LHS ALONG WITH RELATIONAL OPERATORS.

SYNTAX:

LHS RHS

COLUMN\_NAME/EXPRESSION </>/<=/>= ANY(V1,V2,.......);

>IT WORKS ON OR CONDITION.

A:2000 T

B:3000 T

SAL>ANY(1000,2000,3000,4000)

2000>ANY(

3000>ANY(

SELECT \* FROM EMPS WHERE SAL>ANY(SELECT SAL FROM EMPS WHERE JOB='WAITER');

SELECT \* FROM EMPS WHERE SAL>(SELECT MIN(SAL) FROM EMPS WHERE JOB='WAITER');

140. WQTD FNAME OF EMPS WHOSE MANAGER IS NOT AVAILABLE???

A. SELECT FNAME FROM EMPS WHERE MGR IN(SELECT EID FROM EMPS WHERE STATUS!='AVAILABLE');

141.WQTD DETAILS OF EMPLOYEES WHO HAVE DELIVERED ORDERS AND WHOSE MANAGER HAS ALSO DELIVERED AN ORDER???

A. SELECT \* FROM EMPS WHERE EID IN(SELECT EID FROM ORDERS WHERE STATUS='DELIVERED') AND MGR IN(SELECT EID FROM ORDERS WHERE STATUS='DELIVERED');

SELECT \* FROM EMPS WHERE EID IN(SELECT EID FROM ORDERS WHERE STATUS='DELIVERED') AND MGR IN(SELECT EID FROM EMPS WHERE EID IN(SELECT EID FROM ORDERS WHERE STATUS='DELIVERED'));

142. WQTD FNAME OF EMPS WHO ARE REPORTING TO JAHNAVI AND GETTING SALARY LESS THAN MURALI AND LIVING IN KARNATAKA STATE???

A. SELECT FNAME FROM EMPS WHERE MGR IN(SELECT EID FROM EMPS WHERE FNAME='JAHNAVI') AND SAL<(SELECT SAL FROM EMPS WHERE FNAME='MURALI') AND LID IN(SELECT LID FROM LOCATIONS WHERE STATE='KARNATAKA');

JOINS

-----

IT IS USED TO RETRIEVE THE DATA FROM MULTIPLE TABLES SIMULTANEOUSLY.

TYPES OF JOINS

--------------

1. CROSS JOIN/ CARTESIAN JOIN

2. INNER JOIN

3. OUTER JOIN : i. LEFT OUTER JOIN/LEFT JOIN

ii. RIGHT OUTER JOIN/RIGHT JOIN

4. SELF JOIN

5. NATURAL JOIN

1. CROSS JOIN

-------------

IT IS USED TO MERGE THE RECORDS OF ONE TABLE WITH RECORDS ANOTHER TABLE.

ANSI: AMERICAN NATIONAL STANDARD INSTITUTE.

SYNTAX:

SELECT COLUMN\_NAME/EXPRESSION

FROM TABLE\_NAME T1 CROSS JOIN TABLE\_NAME T2;

IF YOU USE CROSS JOIN:

THE TOTAL NUMBER OF COLUMNS IN RESULTANT TABLE=T1+T2

THE TOTAL NUMBER OF RECORDS IN RESULTANT TABLE=T1\*T2

GIRLS G BOYS B

+---+--------+---+ +---+------+

|GID|GNAME |BID| |BID|BNAME |

+---+--------+---+ +---+------+

| 1.|CHINNI | 2 | | 1 |MANJA |

| 2.|MUNNI | 3 | | 2 |CHINNA|

| 3.|MANJULA | 1 | | 3 |MUNNA |

+---+--------+---+ +---+------+

Q. WQTD CARTESIAN PRODUCT FROM GIRLS AND BOYS TABLE?

A. SELECT \* FROM GIRLS G CROSS JOIN BOYS B;

DRAWBACKS

---------

IT WILL OBTAIN MORE NUMBER OF UNMATCHED RECORDS COMPARED TO MATCHED RECORDS.

143. WQTD CARTESIAN PRODUCT FROM EMPS AND LOCATIONS TABLE?

A. SELECT \* FROM EMPS E CROSS JOIN LOCATIONS L;

2. INNER JOIN

-------------

IT IS USED TO RETRIEVE MATCHED RECORDS FROM DIFFERENT TABLES.

SYNTAX:

SELECT COLUMN\_NAME/EXPRESSION

FROM TABLE\_NAME T1 INNER JOIN TABLE\_NAME T2

ON JOIN\_CONDITION;

ON: IT IS A KEYWORD USED TO WRITE JOIN CONDITION.

JOIN\_CONDITION: IT IS USED TO JOIN MULTIPLE TABLES.

+---+-------+----+ +---+-------+

|GID|GNAME | BID| |BID|BNAME |

| 1 |CHINNI | 2| | 1 | MANJA |

| 2 |MUNNI | 3| | 2 | CHINNA|

| 3 |MANJULA| 1| | 3 | MUNNA |

| 4 |PRIYA | 4| +---+-------+

+---+-------+----+

Q. WQTD DETAILS FROM GIRLS AND BOYS TABLE?

A. SELECT \* FROM GIRLS G INNER JOIN BOYS B

ON G.BID=B.BID

144. WQTD DETAILS FROM EMPS AND LOCATIONS TABLE?

A. SELECT \* FROM EMPS E INNER JOIN LOCATIONS L ON E.LID=L.LID;

145. WQTD ITEMS\_NAME AND RESTAURANT NAME IF ITEM NAME IS MASALA DOSA?

A. SELECT M.NAME, R.NAME FROM MENU\_ITEMS M INNER JOIN RESTAURANTS R ON M.RESTAURANT\_ID=R.RESTAURANT\_ID WHERE M.NAME='MASALA DOSA';

146. WQTD AVERAGE RATING FOR EACH RESTAURANTS?

A. SELECT R.NAME, AVG(RATING) FROM RESTAURANTS R INNER JOIN REVIEWS T ON R.RESTAURANT\_ID=T.RESTAURANT\_ID GROUP BY R.NAME;

147. WQTD TOP 2 RESTAURANT NAME WHICH GOT HIGHEST RATING?

A. SELECT R.NAME, T.RATING FROM RESTAURANTS R INNER JOIN REVIEWS T ON R.RESTAURANT\_ID=T.RESTAURANT\_ID ORDER BY T.RATING DESC LIMIT 2;

148. WQTD ORDERS DETAILS AND CUSTOMER NAMES ALONG WITH DELIVERY PERSON NAME?

A. SELECT O.\*, C.NAME, E.FNAME FROM CUSTOMERS C INNER JOIN ORDERS O INNER JOIN EMPS E ON C.ORDER\_ID=O.ORDER\_ID AND O.EID=E.EID;

149. WQTD CUSTOMER NAME WHO MADE THE HIGHEST PAYMENT?

A. SELECT C.NAME, SUM(P.AMOUNT) FROM CUSTOMERS C INNER JOIN ORDERS O INNER JOIN PAYMENTS P ON C.ORDER\_ID=O.ORDER\_ID AND O.ORDER\_ID=P.ORDER\_ID WHERE P.STATUS='COMPLETED' GROUP BY C.NAME ORDER BY SUM(P.AMOUNT) DESC LIMIT 1;

150. WQTD CUSTOMERS WHO ORDERED FROM RESTAURANT IN A SAME CITY?

A. SELECT C.NAME FROM CUSTOMERS C INNER JOIN ORDERS O INNER JOIN RESTAURANTS R INNER JOIN LOCATIONS L ON C.ORDER\_ID=O.ORDER\_ID AND O.RESTAURANT\_ID=R.RESTAURANT\_ID AND C.LID=L.LID WHERE L.CITY = R.CITY;

151. WQTD REVENUE GENERATED BY EACH STATE?

A. SELECT L.STATE, SUM(P.AMOUNT) FROM PAYMENTS P INNER JOIN ORDERS O INNER JOIN CUSTOMERS C INNER JOIN LOCATIONS L ON P.ORDER\_ID=O.ORDER\_ID AND O.ORDER\_ID=C.ORDER\_ID AND C.LID=L.LID WHERE P.STATUS='COMPLETED' GROUP BY L.STATE;

152. WQTD MONTH WISE REVENUE GENERATED IN THE YEAR 2024?

A. SELECT SUM(P.AMOUNT), MONTH(O.ORDER\_DATE) FROM PAYMENTS P INNER JOIN ORDERS O ON P.ORDER\_ID=O.ORDER\_ID WHERE YEAR(O.ORDER\_DATE)=2024 AND P.STATUS='COMPLETED' GROUP BY MONTH(O.ORDER\_DATE);

3. OUTER JOIN

-------------

1. LEFT OUTER JION

------------------

IT IS USED TO JOIN MATCHED AND UNMATCHED RECORDS OF LEFT TABLE.

SYNTAX:

SELECT COLUMN\_NAME/EXPRESSION

FROM TABLE\_NAME T1 LEFT OUTER JOIN TABLE\_NAME T2

ON T1.COLUMN\_NAME=T2.COLUMN\_NAME;

GIRLS G BOYS B

+---+------+----+ +---+------+

|GID|GNAME |BID | |BID|BNAME |

+---+------+----+ +---+------+

|1 |CHINNI|2 | |1 |DINGA |

|2 |SUNDRI|3 | |2 |CHINNA|

|3 |DINGI |1 | |3 |SUNDRA|

|4 |MUNNI |NULL| +---+------+

+---+------+----+

Q. WQTD MATCHED AND UNMATCHED RECORDS FROM GIRLS TABLE?

A. SELECT \* FROM GIRLS G LEFT OUTER JOIN BOYS B ON G.BID=B.BID

OP:

+---+------+----+----+------+

|GID|GNAME |BID |BID |BNAME |

+---+------+----+----+------+

|1 |CHINNI|2 |2 |CHINNA|

|2 |SUNDRI|3 |3 |SUNDRA|

|3 |DINGI |1 |1 |DINGA |

|4 |MUNNI |NULL|NULL|NULL |

+---+------+----+----+------+

153. WQTD MATCHED AND UNMATCHED RECORDS FROM CUSTOMER TABLES?

CUSTOMERS: LEFT

LOCATIONS: RIGHT

A. SELECT \* FROM CUSTOMERS C LEFT OUTER JOIN LOCATIONS L ON C.LID=L.LID;

2. RIGHT OUTER JOIN

-------------------

IT IS USED TO OBTAIN MATCHED AND UNMATCHED RECORDS OF RIGHT TABLE.

SYNTAX:

SELECT COLUMN\_NAME/EXPRESSION

FROM TABLE\_NAME T1 RIGHT OUTER JOIN TABLE\_NAME T2

ON T1.COLUMN\_NAME=T2.COLUMN\_NAME;

GIRLS G BOYS B

+---+------+----+ +---+------+

|GID|GNAME |BID | |BID|BNAME |

+---+------+----+ +---+------+

|1 |CHINNI|2 | |1 |DINGA |

|2 |SUNDRI|3 | |2 |CHINNA|

|3 |DINGI |1 | |3 |SUNDRA|

|4 |MUNNI |NULL| |4 |RAJA |

+---+------+----+ +---+------+

LEFT RIGHT

Q. WQTD MATCHED AND UNMATCHED RECORDS FROM BOYS TABLE?

A. SELECT \* FROM GIRLS G RIGHT OUTER JOIN BOYS B ON G.BID=B.BID;

GIRLS GNAME BID BID BNAME

3 DINGI 1 1 DINGA

1 CHINNI 2 2 CHINNA

2 SUNDRI 3 3 SUNDRA

NULL NULL NULL 4 RAJA

154. WQTD MATCHED AND UNMATCHED RECORDS FROM LOCATIONS TABLE?

A. SELECT \* FROM CUSTOMERS C RIGHT OUTER JOIN LOCATIONS L ON C.LID=L.LID;

155. WQTD CITY WHERE NO CUSTOMERS ARE LIVING?

A. SELECT \* FROM LOCATIONS L LEFT OUTER JOIN CUSTOMERS C ON C.LID=L.LID WHERE C.LID IS NULL;

156. WQTD EMPLOYEE FNAME WHO NEVER HANDLED AN ORDER?

A. SELECT FNAME FROM EMPS E LEFT OUTER JOIN ORDERS O ON E.EID=O.EID WHERE O.EID IS NULL;

157. WQTD RESTAURANTS NAMES WITH NO REVIEWS?

A. SELECT NAME FROM RESTAURANTS R LEFT OUTER JOIN REVIEWS RS ON R.RESTAURANT\_ID=RS.RESTAURANT\_ID WHERE RS.RESTAURANT\_ID IS NULL;

4. SELF JOIN

------------

IT IS USED TO OBTAIN MATCHED RECORDS FROM SAME TABLE.

SYNTAX:

SELECT COLUMN\_NAME/EXPRESSION

FROM TABLE\_NAME T1 JOIN TABLE\_NAME T2

ON JOIN CONDITIONS;

----------------- ----------------

|EID|NAME MGR |EID NAME MGR

|1 |SUNDRA 2 |1 SUNDRA 2

|2 |MUNNA 3 |2 MUNNA 3

|3 |RAJA 4 |3 RAJA 4

|4 |DINGA NULL |4 DINGA NULL

----------------- -----------------

Q. WQTD DETAILS OF EMPS ALONG WITH THEIR MANAGERS FROM EMP TABLE?

A. SELCT \* FROM EMP E1 JOIN EMP E2 ON E1.MGR=E2.EID;

158. WQTD EMPLOYEE FNAME ALONG WITH HIS MANAGER FNAME FROM EMPS TABLE?

A. SELECT E1.FNAME AS 'EMPLOYEE NAME',E2.FNAME AS 'MANAGER NAME' FROM EMPS E1 JOIN EMPS E2 ON E1.MGR=E2.EID;

159. WQTD EMPLOYEE FNAME, SALARY ALONG WITH THEIR MANAGER FNAME, SALARY IF EMP IS GETTING SALARY>35000 AND MGR GETTING SAL <500000?

A. SELECT E1.FNAME AS EMPLOYEE, E1.SAL AS SALARY,E2.FNAME AS MANAGER,E2.SAL AS SALARY FROM EMPS E1 JOIN EMPS E2 ON E1.MGR=E2.EID WHERE E1.SAL>35000 AND E2.SAL<500000;

160. WQTD EMPLOYEE FNAME, DOB ALONG WITH MANAGER FNAME, DOB IF EMP IS ELDER THAN THIS MANAGER?

A. SELECT E1.FNAME AS EMPLOYEE, E1.DOB,E2.FNAME AS MANAGER,E2.DOB FROM EMPS E1 JOIN EMPS E2 ON E1.MGR=E2.EID WHERE E1.DOB<E2.DOB;

161. WQTD EMP FNAME, JOB, MGR FNAME, JOB ALONG WITH THERE CITY NAME IF EMP WORING AS DELIVERY OR CHEF AND EMP CITY IS EITHER DELHI OR JAIPUR AND MANAGER IS WORING AS WAITER?

A. SELECT E1.FNAME AS 'EMPLOYEE NAME', E1.JOB AS 'EMPLOYEE JOB',L1.CITY AS 'EMPLOYEE CITY', E2.FNAME AS 'MANAGER NAME', E2.JOB AS 'MANAGER JOB', L2.CITY AS 'MANAGER CITY' FROM EMPS E1 JOIN EMPS E2 INNER JOIN LOCATIONS L1 INNER JOIN LOCATIONS L2 ON E1.MGR=E2.EID AND E1.LID=L1.LID AND E2.LID=L2.LID WHERE E1.JOB IN('DELIVERY','CHEF') AND L1.CITY IN ('DELHI', 'JAIPUR') AND E2.JOB='WAITER';

162. WQTD FNAME OF AMAN'S MANAGER'S MANAGER?

A. SELECT E3.FNAME FROM EMPS E1 JOIN EMPS E2 JOIN EMPS E3 ON E1.MGR=E2.EID AND E2.MGR=E3.EID WHERE E1.FNAME='AMAN';

163. WQTD DEATAILS OF EMPS WHO ARE GETTING SALARY MORE THAN KARAN?

A. SELECT E1.\* FROM EMPS E1 JOIN EMPS E2 ON E1.SAL>E2.SAL WHERE E2.FNAME='KIRAN';

164. WQTD DETAILS OF EMPS WHO ARE REPORTING TO PRIYA IF PRIYA IS GETTING SAL > KIRAN?

A. SELECT E1.\* FROM EMPS E1 JOIN EMPS E2 JOIN EMPS E3 ON E1.MGR=E2.EID AND E2.SAL>E3.SAL WHERE E2.FNAME='PRIYA' AND E3.FNAME='KIRAN' ;

------------------------------------------

28 JULY 25

NATURAL JOIN

-------------

IT IS USED TO OBTAIN MATCHED RECORDS FROM MULTIPLE TABLES BASED ON COMMEN ATTRIBUTE.

SYNTAX:

SELECT COLUMN\_NAME/EXPRESSION

FROM TABLE\_NAME T1 NATURAL JOIN TABLE\_NAME T2;

SELECT \*

FROM CUSTOMERS E1 NATURAL JOIN LOCATIONS L; MATCHED RECORDS

SELECT \*

FROM EMPS E1 NATURAL JOIN EMPS E2; MATCHED RECORDS

SELECT \*

FROM EMPS E1 NATURAL JOIN MENU\_ITEMS M; merged records (no commen attribute)

SET OPERATORS:

--------------

1.UNION

2.UNION ALL

3.INTERSECTION

A:{1,2,3,4,5} B:{5,6,7}

A UNION B:{1,2,3,4,5,6,7}

A UNION ALL B:{1,2,3,4,5,5,6,7}

A INTERSECTION B:{5}

UNION:

------

IT IS USED TO RETRIEVE THE DATA FROM MULTIPLE TABLE'S VERTICALLY.

>IT WILL AVOID DUPLICATE VALUES FROM THE O/P.

UNION ALL:

----------

IT IS USED TO RETRIVE THE DATA FROM MULTIPLE TABLE'S VERTICALLY

>IT WILL INCLUDE THE DUPLICATE VALUES IN THE O/P.

RULES:

------

1.WE SHOULD USE SEMICOLEN FOR THE LAST QUERY

2.WE SHOULD USE ROUND BRACES FOR THE QUERIES.

3.WE SHOULD USE SAME NUMBER OF COLUMNS IN THE SELECT CLAUSES.

(SELECT FNAME

FROM EMPS)

UNION

(SELECT FNAME

FROM EMPS); O/P:10 FNAME

(SELECT FNAME

FROM EMPS)

UNION ALL

(SELECT FNAME

FROM EMPS); O/P:20 FNAME

165.WQTD JOB,FNAME IN LOWER CASE IF EMPS ARE WORKING AS WAITER ELSE PRINT JOB,FNAME IN REVERSE FORMAT???

(SELECT JOB,LOWER(FNAME) AS FNAME

FROM EMPS

WHERE JOB='WAITER')

UNION

(SELECT JOB,REVERSE(FNAME)

FROM EMPS

WHERE JOB!='WAITER');

166.WQTD MATCHED AND UNMATCHED RECORDS FROM BOTH CUSTOMERS AND LOCATIONS TABLE???

CUSTOMERS:LEFT

LOCATIONS:RIGHT

(SELECT \*

FROM CUSTOMERS C LEFT OUTER JOIN LOCATIONS L

ON C.LID=L.LID)

UNION

(SELECT \*

FROM CUSTOMERS C RIGHT OUTER JOIN LOCATIONS L

ON C.LID=L.LID);

167.WQTD DETAILS OF 4TH AND 7TH RECORD EMPS TABLE??

(SELECT \*

FROM EMPS

LIMIT 1 OFFSET 3)

UNION

(SELECT \*

FROM EMPS

LIMIT 1 OFFSET 6);

168.WQTD DETAILS OF 2ND,3RD,5TH AND 8TH RECORD FROM EMPS TABLE??

(SELECT \*

FROM EMPS

LIMIT 2 OFFSET 1)

UNION

(SELECT \*

FROM EMPS

LIMIT 1 OFFSET 4)

UNION

(SELECT \*

FROM EMPS

LIMIT 1 OFFSET 7);

CASE STATEMENT:

--------------

IT IS USED TO PASS CERTAIN CONDITIONS WHILE DISPLAYING THE RECORDS.

SYNTAX:

SELECT CASE

WHEN CONDITION\_1 THEN VALUE\_1

WHEN CONDITION\_2 THEN VALUE\_2

'

'

WHEN CONDITION\_N THEN VALUE\_1

ELSE DEFAULT\_VALUE

END AS ALIAS\_NAME

FROM TABLE\_NAME;

169.WQTD FNAME AND IF GENDER='M' PRINT IT AS SIGMA\_MALE AND IF GENDER='F' PRINT IT AS DADS LIL PRINCESS ELSE PRINT IT AS OTHER???

ROLE

SELECT FNAME,CASE

WHEN GENDER='M' THEN 'SIGMA MALE'

WHEN GENDER='F' THEN 'DADS LIL PRINCES'

ELSE 'OTHER'

END AS ROLE

FROM EMPS;

170.WQTD FNAME,DOJ AND IF EMPLOYEE HIRED BEFORE 2020 DISPLAY AS SENIOR ELSE JUNIOR?

SELECT FNAME,DOJ,CASE

WHEN YEAR(DOJ)<2020 THEN 'SENIOR'

ELSE 'JUNIOR'

END AS ROLE

FROM EMPS

ORDER BY ROLE ASC;

TO DISABLE AUTOCOMMIT IN MYSQL

------------------------------

SET AUTOCOMMIT=0;

BASICALLY,ALL THE DDL COMMANDS ARE AUTOCOMMIT COMMANDS.

TRANSACTION CONTROL LANGUAGE

----------------------------

1.COMMIT

2.ROLLBACK

3.SAVEPOINT

1.COMMIT

--------

TO SAVE ALL THE TRANSACTIONS PERMENENTLY INSIDE THE DATABASE WE USE COMMIT.

SYNTAX:

COMMIT;

2.ROLLBACK

-----------

TO ROLL OUT THE OPERATIONS UPTO PREVIOUSELY USED COMMIT STATEMENT WE USE ROLLBACK.

SYNTAX:

--------

ROLLBACK;

>WE CAN GETBACK THE DELETED RECORDS BY USING ROLLBACK,IF THE COMMIT IS NOT USED AFTER DELETE OPERATION

>WE CAN ROLL OUT THE OPERATIONS UPTO CERTAIN SAVEPOINTS.

SYNTAX:

-------

ROLLBACK TO SAVEPOINT\_NAME;

SAVEPOINT

---------

TO MARK ONE POSITION BETWEEN THE TRANSACTION WE USE SAVEPOINT.

>HERE, THE DATA WILL BE SAVED TEMPORARILY,BUT NOT PERMENENTLY INSIDE DATABASE.

SYNTAX:

SAVEPOINT SAVEPOINT\_NAME;

DCL(DATA CONTROL LANGUAGE)

--------------------------

IT IS USED TO PROVIDE/GRAND THE PERMISSION OF THE DATA FROM ONE USER TO ANOTHER USER.

SYNTAX:

GRANT SQL\_STATEMENT ON TABLE\_NAME TO 'USER\_NAME'@'HOST\_NAME';

TO VIEW ALL THE USERS PRESENT IN MYSQL

--------------------------------------

STEP1:

USE INFORMATION\_SCHEMA;

STEP2:

SELECT \* FROM USER\_ATTRIBUTES;

TO VIEW ACTIVE USERS PRESENT IN MYSQL

-------------------------------------

SYNTAX:

SELECT USER();

TO CREATE USER IN MYSQL

-----------------------

SYNTAX:

CREATE USER 'USERNAME'@'HOSTNAME' IDENTIFIED BY 'PASSWORD';

HOSTNAME:

USERNAME:PENTAGON

HOSTNAME:LOCALHOST

PASSWORD:SQL

TO ACCESS MYSQL ACCOUNT IN COMMAND PROMT:

-----------------------------------------

SYNTAX:

mysql -u username -p

mysql -u PENTAGON -p

GRANT SELECT ON EMPS TO 'PENTAGON'@'LOCALHOST';

GRANT UPDATE ON EMPS TO 'PENTAGON'@'LOCALHOST';

GRANT ALL ON EMPS TO 'PENTAGON'@'LOCALHOST';

ALL: IT IS USED TO PASS ALL THE PERMISSIONS AT A TIME.

REVOKE

------

IT IS USED TO GETBACK THE PERMISSION OF THE DATA FROM ANOTHER USER.

SYNTAX:

REVOKE SQL\_STATEMENT ON TABLE\_NAME FROM 'USERNAME'@'HOSTNAME';

REVOKE DELETE ON EMPS FROM 'PENTAGON'@'LOCALHOST';

REVOKE ALL ON EMPS FROM 'PENTAGON'@'LOCALHOST';

TO DROP USERS FROM MYSQL

------------------------

SYNTAX:

DROP USER 'USERNAME'@'HOSTNAME';

DROP USER 'PENTAGON'@'LOCALHOST';

1. IS IT POSSIBLE TO CREATE DUPLICATE TABLE?

SYNTAX:

CREATE TABLE TABLE\_NAME(SELECT \* FROM TABLE\_NAME);

EMPS1:EMPS

CREATE TABLE EMPS(SELECT \* FROM EMPS);

2. IS IT POSSIBLE TO CREATE DUPLICATE TABLE WITHOUT RECORDS?

SYNTAX:

CREATE TABLE TABLE\_NAME(SELECT \* FROM TABLE\_NAME WHERE FALSE\_CONDITION);

EMPS2:EMPS

CREATE TABLE EMPS2(SELECT \* FROM EMPS2 WHERE FNAME='SOMETHING'); //GIVE ANY FALSE CONDITION

3. IS IT POSSIBLE TO ADD RECORDS FROM ONE TABLE TO ANOTHER TABLE?

YES...(TABLE STRUCTURE SHOULD BE SAME FOR THE TABLES)

SYNTAX:

INSERT INTO TABLE\_NAME(SELECT STATEMENT);

EMPS2: DELIVERYY BOYS DATA FROM EMPS TABLE

INSERT INTO EMPS2(SELECT \* FROM EMPS WHERE JOB='DELIVERY');

WAITER\_DATA:WAITERS DATA FROM EMPS TABLE

CREATE TABLE WAITER\_DATA(SELECT \* FROM EMPS WHERE JOB='WAITER');

VIEW:

-----

>IT IS A VIRTUAL TABLE.

>IT DOESN'T OCCUPY ANY MEMORY INSIDE THE DATABASE.

>TO OVERCOME THE PROBLEM OF SUBTABLE WE USE VIEW.

SYNTAX:

CREATE VIEW VIEWNAME AS (SELECT STATEMENT);

WAITER:WAITER DATA FROM EMPS

CREATE VIEW WAITER(SELECT \* FROM EMPS WHERE JOB='WAITER');

TO DROP VIEW:

-------------

SYNTAX:

DROP VIEW VIEW\_NAME;

DROP VIEW WAITER;

171. WQTD EMPLOYEE FNAME WHO ARE GETTING SALARY MORE THAN AVG SALARY IN THERE JOB ROLE?

A. SELECT E1.FNAME FROM EMPS E1 WHERE E1.SAL>(SELECT AVG(E2.SAL) FROM EMPS E2 WHERE E1.JOB=E2.JOB);

CO-RELATED SUBQUERY

-------------------

HERE BOTH INNER QUERY AND OUTER QUERY MUTUALLY DEPENDS ON EACH OTHER.

WORKING PRINCIPAL OF CO-RELATED SUBQUERY

----------------------------------------

>OUTER QUERY EXECUTES PARTIALLY.

>INNER QUERY WILL EXECUTE FOR EACH RECORD OF OUTER QUERRY TABLE.

>OUTER QUERY EXECUTES COMPLETLY AND RETURNS FINAL OUTPUT.

172. WQTD EMP FNAME WHO ARE ELDER THAN THERE MANAGER?

A. SELECT E1.FNAME FROM EMPS E1 WHERE E1.DOB<(SELECT E2.DOB FROM EMPS E2 WHERE E1.MGR=E2.EID);

KEY ATTRIBUTE

-------------

ATTRIBUTES WHICH ARE ELIGIBLE TO BECOME PRIMARY KEY WIILL BE KNOWN AS KEY ATTRIBUTE.

NON-KEY ATTRIBUTE

-----------------

ATTRIBUTES WHICH ARE NOT ELIGIBLE TO BECOME PRIMARY KEY WILL BE KNOWN AS NON-KEY ATTRIBUTE.

SUPER KEY ATTRIBUTE

-------------------

ATTRIBUTE OR COMBINATION OF ATTRIBUTES USED TO UNIQUELY IDENTIFIED THE RECORD WILL BE KNOWN AS SUPER KEY ATTRIBUTE.

CANDIDATE KEY ATTRIBUTE

-----------------------

IT IS SMALLEST SUBSET AMONG SUPER KEY ATTRIBUTE.

A TABLE CAN HAVE N NUMBER OF CANDIDATE KEYS.

PRIMARY KEY ATTRIBUTE

---------------------

ATTRIBUTE WHICH IS USED TO UNIQUELY IDENTIFY THE RECORDS WILL BE KNOWN AS PRIMARY KEY ATRIBUTE.

FOREIGN KEY ATTRIBUTE

---------------------

ATTRIBUTE WHICH IS USED TO ESTABLISED THE CONNECTION BETWEEN MULTIPLE TABLES WILL BE KNOWN AS FOREIGN KEY ATTRIBUTE.

COMPOSITE KEY ATTRIBUTE

-----------------------

COMBINATION OF 2 OR MORE ATTRIBUTES AMONG SUPER KEY ATTRIBUTE WILL BE KNOWN AS COMPOSITE KEY ATTRIBUTE.

COMPOUND KEY ATTRIBUTE

----------------------

IF COMPOSITE KEY ATTRIBUTE CONTAINS ATLEAST ONE FOREIGN KEY THAN IT WILL BE KNOWN AS COMPOUND KEY ATTRIBUTE.

PK NN UNIQUE/NN NN UNIQUE FK

+--+-----+---------+--+----------------+---+

|ID|NAME |PHONE |JOB|EMAIL |DNO|

+--+-----+--------+---+----------------+---+

|1 |ROMEO|90101010|JD |ROMEO@GMAIL.COM |10 |

|2 |ROSE |80101011|PD |ROSE@GMAIL.COM |10 |

|3 |JACK |70101010|FD |NULL |20 |

|4 |ROMEO|60101011|JD |NULL |10 |

+--+-----+--------+---+----------------+---+

KA NKA KA NKA NKA NKA

RANKING FUNCTIONS/WINDOW FUNCTIONS

----------------------------------

IT IS USED TO ASSIGN RANKS FOR ALL THE RECORDS IN A TABLE.

SYNTAX:

-------

SELECT RANKING\_FUNCTION() OVER(PARTITION BY COLUMN\_NAME] ORDER BY COLUMN\_NAME ASC/DESC)

FROM TABLE\_NAME;

PARTITION BY

-----------

IT IS USED TO CREATE GROUPS, IT RESETS THE RANK AFTER EACH GROUP.

TYPES OF RANKING FUNCTIONS

--------------------------

1. ROW\_NUMBER()

2. RANK()

3. DENSE\_RANK()

1. ROW\_NUMBER()

---------------

IT IS USED TO ASSIGN UNIQUE RANKS FOR ALL THE RECORDS IN THE TABLE.

SYNTAX:

SELECT ROW\_NUMBER() OVER([PARTITION BY COLUMN\_NAME] ORDER BY ASC/DESC)

FROM TABLE\_NAME;

EX:

SELECT FNAME,JOB,SAL,ROW\_NUMBER() OVER(PARTITION BY JOB ORDER BY SAL DESC) AS 'RANK' FROM EMPS;

SELECT FNAME,JOB,SAL,ROW\_NUMBER() OVER( ORDER BY SAL DESC) AS 'RANK' FROM EMPS;

DRAWBACK

--------

IT WILL ASSIGN UNIQUE RANKS FOR TIED RECORDS.

2. RANK()

---------

IT IS USED TO ASSIGN RANKS FOR ALL THE RECORDS IN A TABLE.

>HERE IT WILL ASSIGN SAME RANK FOR TIED RECORDS.

SYNTAX:

SELECT RANK() OVER([PARTITION BY COLUMN\_NAME] ORDER BY COLUMN\_NAME ASC/DESC)

FROM TABLE\_NAME;

DRAWBACK

--------

HERE, IT WILL ASSIGN SAME RANK FOR TIED RECORDS BUT IT SKIPS NEXT RANKING NUMBERS.

3.DENSE\_RANK()

--------------

IT IS USED TO ASSIGN RANKS FOR ALL THE RECORDS IN A TABLE.

>IT WILL ASSIGN SAME RANK FOR TIED RECORDS ALSO IT REMAINS NEXT RANKING NUMBER IN SEQUENCIAL ORDER.

SYNTAX:

SELECT DENSE\_RANK() OVER([PARTITION BY COLUMN\_NAME] ORDER BY COLUMN\_NAME ASC/DESC)

FROM TABLE\_NAME;

173. WQTD TOP 2 HIGHEST PAID EMPLOYEE FNAME,SAL AND ASSIGN UNIQUE RANKS FOR THE EMPS?

A. SELECT FNAME, SAL, ROW\_NUMBER() OVER (ORDER BY SAL DESC) AS 'RANK' FROM EMPS ORDER BY SAL DESC LIMIT 2;

DEPENDENCY

----------

IF ONE ATTRIBUTE DEPENDS ON ANOTHER ATTRIBUTE THAN THE PROCESS WILL BE KNOWN AS DEPENDENCY.

TYPES OF DEPENDENCY

------------------

1. TOTAL FUNCTIONAL DEPENDENCY

2. PARTIAL FUNCTIONAL DEPENDENCY

3. TRANSITIVE FUNCTION DEPENDENCY

TOTAL FUNCTIONAL DEPENDENCY

--------------------------

IN A RELATION IF ALL THE ATTRIBUTES DEPENDS ON KEY ATTRIBUTES THAN THE PROCESS WILL BE KNOWN AS TOTAL FUNCTIONAL DEPENDENCY.

PARTIAL FUNCTIONAL DEPENDENCY

----------------------------

IN A RELATION IF ONE ATTRIBUTE PARTIALLY DEPENDS ON COMPOSITE KEY ATTRIBUTE THEN THE PROCESS WILL BE KNOWN AS PARTIAL FUNCTIONAL DEPENDENCY.

TRANSITIVE FUNCTIONAL DEPENDENCY

--------------------------------

IN A RELATION IF AN ATTRIBUTE DEPENDS ON NON KEY WHICH IS DIRECTLY DEPENDS ON KEY ATTRIBUTE THEN THE PROCESS WILL BE KNOWN AS TRANSITIVE FUNCTIONAL DEPENDENCY.

REDUNDANCY

----------

THE PROCESS OF REPETATION OF THE DATA WILL BE KNOWN AS REDUNDANCY.

ANAMOLY

-------

THE PROBLEMS OCCUR DUE TO DML OPERATIONS WILL BE KNOWN AS ANAMOLY.

TYPES OF ANAMOLY

----------------

1. INSERT ANAMOLY

2. DELETE ANAMOLY

3. UPDATE ANAMOLY

NORMALISATION

-------------

THE PROCESS OF SPITING LARGER TABLE INTO SMALLER TABLE TO AVOID REDUNDANCY AND ANAMOLY WILL BE KNOWN AS NORMALISATION.

LEVELS- NORMAL FORMS

1NF

---

TABLE SHOULD NOT CONTAIN DUPLICATE RECORDS

A CELL IN A RECORD MUST CONTAIN SINGLE VALUE DATA

2NF

---

TABLE MUST FOLLOW 1NF

TABLE SHOULD NOT CONTAIN PARTIAL FUNCTIONAL DEPENDENCY

3NF

---

TABLE MUST FOLLOW 2NF

TABLE SHOULD NOT CONTAIN TRANSITIVE FD

THE END

----------------------------------------------------------------------------------------------------------