SQL INTERVIEW QUESTIONS AND ANSWERS

Q1. HOW TO GET UNIQUE VALUES OF A COLUMN?

ANS

SELECT DISTINCT COLUMN NAME FROM TABLE_NAME;

Q2. WHAT ARE THE SEQUENCE OF CLAUSES IN SQL?

ANS

1. WHERE
2. GROUP BY
3. HAVING
4. ORDER BY

Q3. WHAT IS THE DIFFERENCE BETWEEN WHERE AND HACING CLAUSE? CAN WE USE BOTH IN THE SAME QUERY?

ANS

WHERE AND HAVING CLAUSE BOTH ARE CONDITIONAL CLAUSE
WHERE CLAUSE GOES CONDITIONAL ON A COLUMN WITHOUT AGGREGATE FUNCTION
HAVING CLAUSE GOES CONDITIONAL ON A COLUMN WITH AGGREGATE FUNCTION AND GROUP BY CLAUSE
YES, BOTH WHERE AND HAVING CLAUSE CAN BE USED IN THE SAME QUERY

EX:

```
SELECT
COMPANY,
GENDER,
FORMAT(SUM(SPENT_AMOUNT),'C0') AS SPENT
FROM MED_2021
WHERE COMPANY IN ('APPOLO','CIPLA','RELEGARE')
GROUP BY COMPANY,GENDER
HAVING SUM(SPENT_AMOUNT) > 80000
ORDER BY 1,2;
```

Q4. WHAT IS THE DIFFERENCE BETWEEN PRIMARY KEY AND UNIQUE KEY?

ANS

PRIMARY KEY AND UNIQUE KEY BOTH ARE UNIQUE IN NATURE
HOWEVER, UNIQUE CAN HAVE A NULL VALUE

Q5. TYPES OF INDEX IN SQL QUERIES?

ANS

CLUSTERD INDEX
NON-CLUSTERED INDEX
UNIQUE INDEX
FILTERED INDEX
COLUMNSTORE INDEX
HASH INDEX

Q6. HOW TO DESCRIBE A TABLE IN SQL?

ANS

```
SELECT * FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME='MED_2021';
```

Q7. HOW TO SELECT RANDOM RECORDS FROM A TABLE?

ANS

```
SELECT TOP 100 * FROM MED_RANDOM ORDER BY NEWID();
```

Q8. TYPES OF OPERATORS IN SQL QUERIES?

ANS

1. LOGICAL OPERATOR (AND,OR,IN)
2. COMPARISON OPERATOR (=,>,<,>=,<=,<>)
3. SPECIAL OPERATOR (BETWEEN AND, LIKE)
4. ARITHMATIC OPERATOR (+,-,*,/)

Q9. QUESTION ON CASE AND WHEN STATEMENT?

ANS

```
SELECT
CUSTOMER_ID,
COMPANY,
GENDER,
AGE,
CASE
        WHEN AGE < 30 THEN 'LESS THAN 30'
        WHEN AGE >= 30 AND AGE <=50 THEN '30-50'
        WHEN AGE > 50 AND AGE <=70 THEN '51-70'
        ELSE '70+'
END AS AGE_BUCKET,
STATE_CODE,
SPENT_AMOUNT,
CASE
        WHEN SPENT_AMOUNT < 300 THEN 'LESS THAN 300'
        WHEN SPENT_AMOUNT >= 300 AND SPENT_AMOUNT <= 500 THEN '300-500'
        WHEN SPENT_AMOUNT > 500 AND SPENT_AMOUNT <= 1000 THEN '501-1000'
        WHEN SPENT_AMOUNT > 1000 AND SPENT_AMOUNT <= 1500 THEN '1001-1500'
        ELSE '1500+'
END AS SPENT_BUCKET
INTO MED_SELECTION1
FROM MED_2021;
```

Q10. QUESTIONS ON NESTED CASE AND WHEN STATEMENT

ANS

```
SELECT
PRODUCT,
CITY,
PRICE,
UNITS,
CASE
        WHEN PRODUCT='APPLE' THEN
             CASE
                     WHEN CITY='BANGALORE' THEN .20*PRICE
                         WHEN CITY='DELHI' THEN .10*PRICE
                 ELSE 0
                 END
        WHEN PRODUCT='DELL' THEN
             CASE
                     WHEN CITY='BHUBANESWAR' THEN .30*PRICE
                         WHEN CITY='CHENAI' THEN .40*PRICE
                ELSE 0
                END
        WHEN PRODUCT='HP' THEN
             CASE
                     WHEN CITY='MUMBAI' THEN .20*PRICE
                         WHEN CITY='CHENAI' THEN .30*PRICE
                ELSE 0
                END
        ELSE NULL
END AS DISCOUNT
FROM PROD_SALES_IND;
```

Q11. HOW TO TRANSPOSE TABLE IN SQL QUERIES

ANS

PIVOT FUNCTION TO CONVERT ROWS INTO COLUMNS AND **UNPIVOT FUNCTION** TO CONVERT COLUMNS INTO ROWS

EX:

CONVERTING ROWS INTO COLUMNS

```
SELECT
STATE_CODE,
COMPANY,
FEMALE,
MALE,
UNISEX
FROM MED_SUMMARY_V1
PIVOT(SUM(SUBS) FOR GENDER IN (FEMALE,MALE,UNISEX)) AS X;
```

CONVERTING COLUMNS INTO ROWS

```
SELECT
STATE_CODE,
COMPANY,
GENDER,
SUBS
FROM MED_SUMMARY_V2
UNPIVOT (SUBS FOR GENDER IN (FEMALE,MALE,UNISEX)) AS X;
```

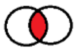
Q12. EXPLAIN JOINS?

ANS

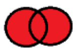
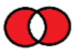
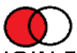
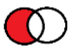
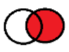
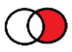
JOINS ARE 2 TYPES

1. VERTICAL JOIN (APPENDING TABLES)
2. HORIZONTAL JOIN (MERGING TABLES)

VERTICAL JOIN- WE USE UNION ALL OR UNION QUERY
HORIZONTAL JOIN- WE USE JOININGS LIKE

INNER JOIN- 

OUTER JOIN-

- FULL OUTER JOIN 
  - UN-MATCHED JOIN FROM FULL JOIN QUERY 
- LEFT OUTER JOIN 
  - LEFT NULL JOIN FROM LEFT JOIN QUERY 
- RIGHT OUTER JOIN 
  - RIGHT NULL JOIN FROM RIGHT JOIN QUERY 

Q13. DIFFERENCE BETWEEN UNION AND UNION ALL QUERY?

ANS

UNION QUERY APPENDS TABLES WITH UNQIUE VALUES
UNION ALL QUERY APPENDS TABLES WITH IRRESPECTIVE OF DUPLICATE VALUES

Q14.IF INNER JOIN IS NOT WORKING IN SQL HOW TO GET INNER JOIN VALUES?

ANS

SOLUTION-1

```
SELECT
A.STU_ID,
A.STU_NAME,
A.GENDER,
A.EDUCATION,
B.STU_ID,
B.YOE,
B.COMPANY,
B.SALARY
FROM STU_EDUCATION AS A
FULL JOIN
STU_EXPERIENCE AS B
ON
A.STU_ID=B.STU_ID
WHERE A.STU_ID IS NOT NULL AND B.STU_ID IS NOT NULL;
```

SOLUTION-2

```
SELECT
A.STU_ID,
A.STU_NAME,
A.GENDER,
A.EDUCATION,
B.STU_ID,
B.YOE,
B.COMPANY,
B.SALARY
FROM STU_EDUCATION AS A
LEFT JOIN
STU_EXPERIENCE AS B
ON
A.STU_ID=B.STU_ID
WHERE B.STU_ID IS NOT NULL;
```

SOLUTION-3

```
SELECT
A.STU_ID,
A.STU_NAME,
A.GENDER,
A.EDUCATION,
B.STU_ID,
B.YOE,
B.COMPANY,
B.SALARY
FROM STU_EDUCATION AS A
RIGHT JOIN
STU_EXPERIENCE AS B
ON
A.STU_ID=B.STU_ID
WHERE A.STU_ID IS NOT NULL;
```

SOLUTION-4

```
SELECT
A.STU_ID,
A.STU_NAME,
A.GENDER,
A.EDUCATION,
B.STU_ID,
B.YOE,
B.COMPANY,
B.SALARY
FROM STU_EDUCATION AS A
LEFT JOIN
STU_EXPERIENCE AS B
ON
A.STU_ID=B.STU_ID

INTERSECT

SELECT
A.STU_ID,
A.STU_NAME,
A.GENDER,
A.EDUCATION,
B.STU_ID,
B.YOE,
B.COMPANY,
B.SALARY
FROM STU_EDUCATION AS A
RIGHT JOIN
STU_EXPERIENCE AS B
ON
A.STU_ID=B.STU_ID;
```

Q15. HOW MANY RECORDS WIL RGENERATE IF WE JOIN THESE TWO TABLES WITH ID?

ANS

TABLE-1
ID
1
1
1
2
2

TABLE-2
ID
1
1
2

INNER JOIN- 8
FULL JOIN- 8
UN-MATCHED- 0
LEFT JOIN- 8
LEFT NULL JOIN- 0
RIGHT JOIN- 8
RIGHT NULL JOIN- 0

Q16. HOW MANY WAYS WE CAN REMOVE DUPLICATE VALUES IN SQL QURIES?

ANS

1. USING PROC SQL DISTINCT CLUASE
2. PROC SQL WITH GROUP BY AND COUNT, HAVING COUNT > 1
3. ORDER DATA AND GERRING ROW_NUMBER TO SELECT FIRST ROW AS UNIQUE VALUES
4. USING ALL COLUMN NAME AND GROUP BY WITH ALL COLUMN NAME

Q17. HAVE YOU DESIGNED ANY DATA MODEL? IF YES, WHAT IS YOUR APPROACH?

ANS

BEFORE DESIGNING DATA MODEL, WE MUST IDENTIFY THE FACT AND DIMENSION TABLES
THEN UNDERSTAND THE BEST DATA MODEL TO FIT IN LIKE

- STAR SCHEMA
- SNOWFLAKES

Q18. HAVE YOU EVER FACED CHALLENGES DEISGNING DATA MODEL? IF YES, PLEASE EXPLAIN?

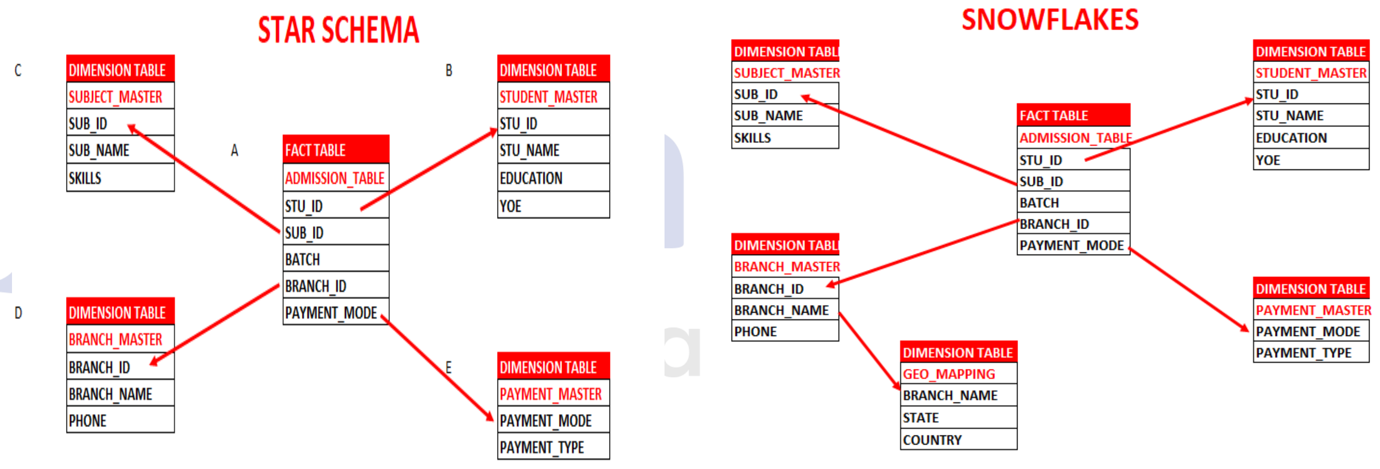
ANS

COMPLEXITY IN DATA MODEL DESIGN COMES, WHERE MULIPLE SOURCE FILES TO BE MAPPED AND SOURCE FILES INFORMATION IS NOT IN STRUCTURED FORM. SO SOMETIMES THIS OUTPUT IN MANY TO MANY RELATIONSHIP VALUES AND IMPACT IN DUPLICATE RECORDS

SO BEFORE DESIGNING THE DATA MODEL WE CHECK DATA STRUCTURE FIRST, THEN VALUES AND RELATIONSHIP AND THE TYPE OF DATA MODEL IS GOING DESIGN FOR

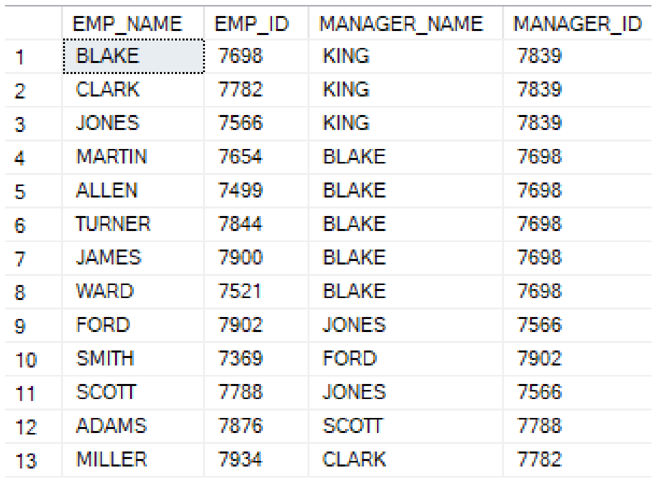
Q19. IDENTIFY THE DATA MODEL?

ANS



Q20. HOW TO FIND EMPLOYEE TO EMPLOYEE MANAGER AND MANAGERS MANAGER?

ANS



	EMP_NAME	EMP_ID	MANAGER_NAME	MANAGER_ID
1	BLAKE	7698	KING	7839
2	CLARK	7782	KING	7839
3	JONES	7566	KING	7839
4	MARTIN	7654	BLAKE	7698
5	ALLEN	7499	BLAKE	7698
6	TURNER	7844	BLAKE	7698
7	JAMES	7900	BLAKE	7698
8	WARD	7521	BLAKE	7698
9	FORD	7902	JONES	7566
10	SMITH	7369	FORD	7902
11	SCOTT	7788	JONES	7566
12	ADAMS	7876	SCOTT	7788
13	MILLER	7934	CLARK	7782

```
SELECT
A.EMP_NAME,
A.EMP_ID,
A.MANAGER_NAME AS MANAGER_NAME,
B.MANAGER_NAME AS MANAGERS_MANAGER_NAME,
B.MANAGER_ID AS MANAGERS_MANAGER_ID
FROM EMPLOYEE_MANAGER_TBL AS A
LEFT JOIN
EMPLOYEE_MANAGER_TBL AS B
ON
A.MANAGER_ID=B.EMP_ID;
```

Q21. EXAPLIN FOLLOWING FUNCTIONS IN DETAIL?

ANS

1. SUBSTRING - `SUBSTRING(NAME,1,CHARINDEX(' ',NAME)) AS FIRST_NAME`
2. CHARINDEX ----`CHARINDEX(' ',NAME)`
3. PATINDEX---- `PATINDEX('% %',NAME)`
4. REPLACE---- `REPLACE('SACHIN ENDULKAR','ENDULKAR','TENDULKAR')`

Q22. HOW TO GET DATE DIFFERENCE IN YEAR, MONTHS, DAYS?

ANS

USING **DATEDIFF** FUNCTION

```
DATEDIFF(YEAR,'2019-FEB-21',GETDATE()) AS YEARS_GAP
DATEDIFF(MONTH,'2019-FEB-21',GETDATE()) AS MONTHS_GAP
DATEDIFF(DAY,'2019-FEB-21',GETDATE()) AS DAYS_GAP
```

Q23. HOW TO ADD DATE FIELD BY YEAR, MONTHS, DAYS?

ANS

```
DATEADD(YEAR,2, GETDATE()) AS YEARS_ADD
DATEADD(MONTH,2, GETDATE()) AS MONTHS_ADD
DATEADD(DAY,2, GETDATE()) AS DAYS_ADD
```

Q24. HOW TO CONVERT DATA TYPES IN SQL?

ANS

```
SELECT CAST(123 AS VARCHAR);
SELECT CAST('02-27-2020' AS DATE);

SELECT CONVERT(VARCHAR,123 );
SELECT CONVERT(DATE,'02-27-2020');
```

Q25. WHAT IS THE USER DEFINED FUNCTIONS IN SQL?

ANS

1. SCALAR FUNCTIONS
2. TABLE FUNCTIONS

Q26. WHAT IS THE USE OF SUB-QUERIES? CAN WE USE ORDER BY IN SUB-QUERY?

ANS

A SUBQUERY IS USED TO RETURN DATA THAT WILL BE USED IN THE MAIN QUERY AS A CONDITION TO FURTHER RESTRICT THE DATA TO BE RETRIEVED

WE CAN NOT USE ORDER BY CLAUSE IN SUB-QUERIES

Q27. WHAT ARE THE NUMERIC FORMATS WE HAVE?

ANS

```
SELECT FORMAT(12675545234,'#;(#);[zero]') AS SALES;
SELECT FORMAT(12675545234,'G0') AS SALES;
SELECT FORMAT(12675545234,'0,#.00') AS SALES;
SELECT FORMAT(12675545234,'N2') AS SALES;
SELECT FORMAT(12675545234,'0,#.') AS SALES;
SELECT FORMAT(12675545234,'N0') AS SALES;

SELECT FORMAT(100,'C0') AS SALES;
SELECT FORMAT(100,'C2') AS SALES;
SELECT FORMAT(100,'C0','fr-FR') AS SALES;
SELECT FORMAT(100,'C0','zh-cn') AS SALES;
SELECT FORMAT(100,'C0','de-DE') AS SALES;
```

Q28. WHAT IS VIEW? CAN WE USE ORDER BY CLAUSE IN VIEW?

ANS

A VIEW IS THE RESULT SET OF A STORED QUERY ON THE DATA, WHICH THE DATABASE USERS CAN QUERY JUST AS THEY WOULD IN A PERSISTENT DATABASE COLLECTION OBJECT. THIS PRE-ESTABLISHED QUERY COMMAND IS KEPT IN THE DATABASE DICTIONARY

WE CAN NOT USE ORDER BY CLAUSE IN VIEW
VIEW CAN BE DROPPED AND REPLACED

```
CREATE VIEW MED_SUMMARY_VIEW
AS
SELECT
STATE_CODE,
COMPANY,
GENDER,
COUNT(CUSTOMER_ID) AS SUBS,
SUM(NO_OF_TRIPS) AS VISITS,
SUM(SPENT_AMOUNT) AS SPENT
FROM MED_2021
GROUP BY STATE_CODE,COMPANY,GENDER;

DROP VIEW MED_SUMMARY_VIEW;

CREATE OR REPLACE VIEW MED_SUMMARY_VIEW AS
SELECT
STATE_CODE,
COMPANY,
COUNT(CUSTOMER_ID) AS SUBS,
SUM(NO_OF_TRIPS) AS VISITS,
SUM(SPENT_AMOUNT) AS SPENT
FROM MED_2021
GROUP BY STATE_CODE,COMPANY;
```

Q29. WHAT IS STORE PROCEDURE?

ANS

A STORED PROCEDURE IS A PREPARED SQL CODE THAT YOU CAN SAVE, SO THE CODE CAN BE REUSED OVER AND OVER AGAIN. SO IF YOU HAVE AN SQL QUERY THAT YOU WRITE OVER AND OVER AGAIN, SAVE IT AS A STORED PROCEDURE, AND THEN JUST CALL IT TO EXECUTE IT

```
EX:
CREATE PROC MED_PROC
--PARAMETERS
@COMP CHAR(15),
@GEN CHAR(15),
@AG INT,
@STCD CHAR(15)

AS
BEGIN

--PASS PARAMTERS INTO THE PROHRAM

SELECT CUSTOMER_ID,COMPANY,GENDER,AGE,STATE_CODE,SPENT_AMOUNT
FROM MED_2021
WHERE COMPANY=@COMP AND GENDER=@GEN AND AGE > @AG AND STATE_CODE=@STCD

END;

EXEC MED_PROC 'APPOLO','FEMALE',45,'QLD';
EXEC MED_PROC 'CIPLA','MALE',65,'NSW';

DROP PROC MED_PROC;
```

INTERVIEW PRACTICALLY ASKED QUESTIONS AND SOLVING APPROACHES

Q1. GIVEN PROD_SALES TABLE FIND THE ANSWER TO QUESTIONS GIVEN BELOW?

1. GET SALES MONTH OVER MONTH
2. GET CUSTOMER LEVEL TOP 3 PRODUCTS SELLING

CUSTOMER_ID	PRODUCT_ID	SALES_DATE	SALES_AMOUNT
CUST4	PROD1	23-09-2020	234
CUST10	PROD7	20-01-2020	422
CUST5	PROD7	14-01-2020	759
CUST1	PROD1	18-03-2020	880
CUST4	PROD5	08-01-2020	386
CUST9	PROD5	23-02-2020	527
CUST8	PROD8	24-11-2020	915
CUST2	PROD6	05-12-2020	270
CUST1	PROD10	05-05-2020	578
CUST1	PROD9	06-10-2020	233
CUST10	PROD10	16-10-2020	872
CUST7	PROD6	16-04-2020	153
CUST2	PROD7	17-03-2020	628
CUST2	PROD7	14-07-2020	789
CUST8	PROD1	10-10-2020	664
CUST10	PROD6	21-12-2020	167
CUST2	PROD2	08-07-2020	888
CUST8	PROD4	26-07-2020	279
CUST3	PROD4	22-09-2020	780
CUST5	PROD10	16-01-2020	178
CUST4	PROD10	10-12-2020	141
CUST5	PROD9	20-03-2020	414
CUST4	PROD5	04-07-2020	419
CUST7	PROD6	08-10-2020	298
CUST6	PROD2	26-10-2020	439
CUST3	PROD2	14-04-2020	326
CUST3	PROD7	11-08-2020	806
CUST2	PROD1	24-08-2020	514
CUST8	PROD9	03-01-2020	412
CUST4	PROD4	01-05-2020	649

ANS

1. GET SALES MONTH OVER MONTH

```
--STEP-1    CREATE MONTH FIELD FROM SALES DATE FIRST AND STORE TO A NEW TABLE

SELECT
CUSTOMER_ID,
PRODUCT_ID,
SALES_DATE,
DATENAME(MONTH,SALES_DATE) AS SALES_MONTH,
SALES_AMOUNT
INTO PROD_SALES_V1
FROM PROD_SALES;
```

```
--STEP-2    SUMMARIZE DATA BY MONTH WISE SALES

SELECT
SALES_MONTH,
SUM(SALES_AMOUNT) AS TOTAL_SALES
FROM PROD_SALES_V1
GROUP BY SALES_MONTH;
```

ANS

2. GET CUSTOMER LEVEL TOP 3 PRODUCTS SELLING

```
--STEP-1   SUMMARIZE DATA

SELECT
CUSTOMER_ID,
PRODUCT_ID,
SUM(SALES_AMOUNT) AS TOTAL_SALES
INTO PROD_SALES_V2
FROM PROD_SALES
GROUP BY CUSTOMER_ID,PRODUCT_ID;

SELECT * FROM PROD_SALES_V2;


--STEP-2    GET ORDERING SALES IN DESCENDING AND RANK IT

SELECT
CUSTOMER_ID,
PRODUCT_ID,
TOTAL_SALES,
DENSE_RANK() OVER (PARTITION BY CUSTOMER_ID ORDER BY TOTAL_SALES DESC) AS RANKING
INTO PROD_SALES_V3
FROM PROD_SALES_V2;

--STEP-3    SELECT TOP 3 RANKING

SELECT * FROM PROD_SALES_V3 WHERE RANKING <=3;
```

Q2. GIVEN RESTAURANT_TRANSACTION TABLE FIND THE ANSWER TO QUESTIONS GIVEN BELOW?

1. GET RESTAURANT WISE MONTH ON MONTH SALES
2. GET EACH RESTAURANT WISE TOP 5 CUSTOMERS BY SPENT
3. GET EACH RESTAURANT WISE AVERAGE DAYS GAP BY CUETOMER VISITS
4. HOW TO GET CUSTOMER INSIGHTS FROM THIS DATA

REST_NAME	CUST_NAME	VISIT_DATE	SPENT_AMOUNT
KFC	Rick Hansen	15-03-2019	2343
MACD	Justin Ritter	10-11-2019	565
GURU	Craig Reiter	01-08-2019	1123
PURVI	Katherine Murray	26-07-2019	2032
DALMA	Jim Mitchum	17-03-2019	4601
MACD	Toby Swindell	20-08-2019	6704
GREEN	Mick Brown	22-08-2019	360
PURVI	Jane Waco	01-02-2019	2304
MALWA	Joseph Holt	20-12-2019	4552
KITTO	Greg Maxwell	08-03-2019	3456
GYMNI	Anthony Jacobs	12-10-2019	3284
JUNGLE	Mick Brown	09-12-2019	6650
TARINI	Magdelene Morse	20-05-2019	5777
DALMA	Craig Reiter	23-03-2019	1109
ODIA	Vicky Freymann	13-08-2019	2624
DINE-IN	Craig Reiter	11-06-2019	6511
LOGGO	Greg Maxwell	26-12-2019	2170
REFORM	Jim Mitchum	24-01-2019	3100
GYMNI	Peter Fuller	14-02-2019	4741
MALWA	Ben Peterman	04-03-2019	3144
KFC	Thomas Boland	07-12-2019	6197
GURU	Rick Hansen	17-03-2019	4996
HOWDI	Magdelene Morse	20-02-2019	3302
LOGGO	Patrick Jones	18-09-2019	2648
DALMA	Jim Sink	09-11-2019	857
ODIA	Patrick Jones	08-02-2019	6721
MACD	Ritsa Hightower	13-08-2019	819
YO CHOW	Ann Blume	26-09-2019	5643
KFC	Rick Hansen	04-03-2019	3870
LEONE	Sue Ann Reed	09-01-2019	4817
MACD	Ann Blume	15-08-2019	2790
YO CHOW	Jason Klamczynski	16-12-2019	3387
MAYURI	Laurel Beltran	01-01-2019	2886

ANS

1. GET RESTAURANT WISE MONTH ON MONTH SALES

```
--STEP-1   CREATE MONTH FIELD FIRST AND STORE TO A NEW TABLE

SELECT
REST_NAME,
CUST_NAME,
VISIT_DATE,
DATENAME(MONTH,VISIT_DATE) AS VISITS_MONTH,
SPENT_AMOUNT
INTO RESTAURANT_TRANSACTIONS_V1
FROM
RESTAURANT_TRANSACTIONS;

SELECT * FROM RESTAURANT_TRANSACTIONS_V1;

--STEP-2   SUMMARIZE DATA BY REST_NAME AND MONTH WISE

SELECT
REST_NAME,
VISITS_MONTH,
SUM(SPENT_AMOUNT) AS TOTAL_SPENT
FROM RESTAURANT_TRANSACTIONS_V1
GROUP BY REST_NAME,VISITS_MONTH
ORDER BY 1,2;
```

ANS

2. GET EACH RESTAURANT WISE TOP 5 CUSTOMERS BY SPENT

```
--STEP-1     SUMMARIZE DATA BY REST_NAME AND CUST_NAME

SELECT
REST_NAME,
CUST_NAME,
SUM(SPENT_AMOUNT) AS TOTAL_SPENT
INTO RESTAURANT_TRANSACTIONS_V2
FROM RESTAURANT_TRANSACTIONS
GROUP BY REST_NAME,CUST_NAME;


--STEP-2       CREATING RANKING BASED ON REST_NAME AND SPENT ON DESCENDING ORDER

SELECT
REST_NAME,
CUST_NAME,
TOTAL_SPENT,
DENSE_RANK() OVER (PARTITION BY REST_NAME ORDER BY TOTAL_SPENT DESC) AS RANKING
INTO RESTAURANT_TRANSACTIONS_V3
FROM RESTAURANT_TRANSACTIONS_V2;

--STEP-3   SELECT TOP 5 CUSTOMERS BY RESTAURANT NAME

SELECT * FROM RESTAURANT_TRANSACTIONS_V3 WHERE RANKING <=5;
```

ANS

3. GET EACH RESTAURANT WISE AVERAGE DAYS GAP BY CUETOMER VISITS

```
--STEP-1 USE LAG FUNCTION TO GET PREVIOUS_DAY VISIT AND USE DATEDIFF FUNCTION TO GET DAYS GAP
BETWEEN PREVIOUS DAY VISIT TO NEXT DAY VISITS DAYS GAP

SELECT
REST_NAME,
CUST_NAME,
VISIT_DATE,
LAG(VISIT_DATE,1) OVER (PARTITION BY REST_NAME,CUST_NAME ORDER BY VISIT_DATE) AS PREV_VISIT_DATE,
DATEDIFF(DAY,LAG(VISIT_DATE,1) OVER (PARTITION BY REST_NAME,CUST_NAME ORDER BY
VISIT_DATE),VISIT_DATE) AS DAY_GAP,
SPENT_AMOUNT
INTO RESTAURANT_TRANSACTIONS_NEW
FROM RESTAURANT_TRANSACTIONS;

--STEP-2  SUMMARIZE DATA BY REST_NAME,CUST_NAME WISE AVERAGE DAYS_GAP IN VISIT

SELECT
REST_NAME,
CUST_NAME,
AVG(DAY_GAP) AS AVG_VISIT_DAYS_GAP
FROM RESTAURANT_TRANSACTIONS_NEW
GROUP BY REST_NAME,CUST_NAME
ORDER BY 1,2;
```

ANS

4. HOW TO GET CUSTOMER INSIGHTS FROM THIS DATA

APPROACH-1

GET RESTAURANT WISE CUSTOMERS TOTAL SEPNT AND FIND HOW MANY CUSTOMERS SPENT ABOVE AVERAGE AND BELOW AVERAGE TO RESTAURANT AVERAGE SPENT

SPENT ABOVE AVERAGE CALL THEM PREMIER
SPENT BELOW AVERAGE CALL THEM NON-PREMIER

APPROACH-2

GET RESTAURANT WISE CUSTOMERS AVG_DAYS_ GAP IN VISITS AND THEIR TOTAL_SPENT
GET RESTARANT LEVEL AVG VISITS DAYS GAP AND AVG SPENT

SEGMENT CUSTOMER WHO ARE BELOW AVG DAYS GAP AND ABOVE AVG SPENT AT RESTAURANT LEVEL AS **PREMIER**
SEGMENT CUSTOMER WHO ARE ABOVE AVG DAYS GAP AND BELOW AVG SPENT AT RESTAURANT LEVEL AS **NON-PREMIER**

THIS HELPS IN CRACKING SQL BASE INTERVIEW QUESTIONS.

MAKE USE OF IT