

2. What is SQL? What are different sublanguages of SQL.

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in relational database. SQL is the standard language for relational database system.

Sub Languages of SQL OR Components of SQL.

1) Data Definition Language (DDL): It is used to create, remove or alter the structure of tables and database objects.

- Creation of table
- Altering table
- Dropping table
- Truncating table

2) Data Retrieval Language (DRL): It enables users to retrieve the data which is used to query the existing database. It includes the following statement

- Insert - Inserting a record into the table.
- Update

3) Data Manipulation Language (DML): It enables users to retrieve, insert, update and delete the data. It is used to query and manipulate the existing database. It includes the following statement.

- Insert - Inserting a record into the table
- Update - Editing & modifying data stored in table
- Delete - Deleting unnecessary records.

4) Data Control Language (DCL): It enables user to control the way data can be accessed or shared. It includes the following two statements.

- Grant - Allows other user to share table
- Revoke - Denying other user to share after granted.

5) Transaction Control Language (TCL): It enables the to save or undo the DML operation on the database

- Commit - saving all DML operation permanently
- Rollback - To undo all DML operation up to last save
- Savepoint - point of the transaction.

Q2. What is Constraints? What are the different types of Constraints.

Constraints consists is a set of rules have to be enforced on data stored. It super control the data being entered in a table for permanent storage.

Types of data Constraints:-

There are two types of data constraints that can be applied data being inserted into an oracle table.

I/O Constraint:

The Input/Output data constraints are further divided into two in oracle table. different constraints.

1) The primary key constraints:

A primary key is one or more column in a table uniquely identify each row in a table. Name of the field that is part of primary key can not contain null value. A primary key column in a table has special attributes.

i) It defines the column as a mandatory column (i.e. the cannot be left blank). As the NOT NULL attribute is active.

2) The Foreign (self Reference) constraint:-

Foreign keys represent relationship between tables. A foreign key is a column (as a group of column) whose values are derived from the primary key or unique key of some other table.

The table in which the foreign key is defined is called a foreign table. The table that defines the primary or unique key and is referred as primary key ~~using~~ the foreign key is called the primary table or master table.

3. What is Database object? Write note on sequences and synonyms.

A sequence is a database object, which can generate unique sequential integer values. It can be used to automatically generate primary key values. A sequence can be either in an ascending or a descending order.

Tables are the basic unit of data storage in an oracle database. Database tables hold all user-accessible data. Each table rows and columns.

- 1) Table
- 2) Views
- 3) Synonyms
- 4) Sequences
- 5) Index
- 6) Cluster
- 7) Function
- 8) Procedure
- 9) Triggers.

Synonyms:- A synonym is an alias for any data, view, sequences, procedure, function, packages, type. Synonyms is family or alias, it requires no storage other than its definition in the data dictionary.

Sequences:- Sequences are database objects from which multiple users can generate unique integers. We can use sequences to automatically generate primary key values.

4. What is views? Explain the purpose of view? Write the syntax to create view.

A view is a tailored presentation of data contained in one or more tables. A view takes the output of a query and a table. Therefore a view can be thought of as a stored query of a 'virtual table'. The table upon which a view is based is called base table.

The advantages of view are as follow:-

- 1) They provides additional level of table security.
- 2) They hide data complexity. The view hides the fact that that this information actually originates from several tables.
- 3) They Simplify Commands for user.
- 4) View provides data in a different perspective than of a base table.

Creating Views:-

By Using Create statement we can create views. Following two Syntax can be used to create views:

Syntax :- Create view <view-name> as select columnname1, columnname2 from <tablename> where columnname = Expression list;

Syntax :- Create view <view.....> as

<query> [with [check option] [read only] [constraint]];

Example :- Create view ven-view as select * from vendor_master;

The above command will create a view with name ven-view which will have the same structure as the vendor_master table & all the rows of base table are accessible through this view.

5. What are different operators used in SQL?

An operator is a reserved word or a character used primarily in an SQL statement's WHERE clause to perform (s), such as comparisons and arithmetic operations. These operators are used to specify conditions in an SQL statement and to serve as conjunctions for multiple conditions in statement.

SQL Arithmetic Operators:-

Example : Assume variable $a=10$ variable $b=20$, the values of arithmetic arithmetic is as follows:-

Operator	Description	Example
$+$ (Addition)	Adds values on either side of the operator	$a+b$ will give 30
$-$ (subtraction)	Subtracts right hand operand from left hand operand.	$a-b$ will give -10
$*$ (multiplication)	Multiplication values on either side of the operator.	$a*b$ will give 200
$/$ (division)	Divides left hand operand by right hand operand	b/a will give 2
$\%$ (modulus)	Divides left hand operand by right hand operand and returns its remainder.	$b \% a$ will give 0

6. What are different datatypes used in PL/SQL.

Sr
No

Category & Description

1. Scalar - Single values with no internal components, such as a NUMBER, DATE or BOOLEAN.
2. Large Object (LOB) - Pointers to objects that are stored separately from other data items, such as text graphic images, video clips and sound waveforms.
3. Composite - Data items that have internal components that can be accessed individually. For example, collections and records.
4. Reference - Pointers to other data items.

7. What is Exception? What are user defined Exception?

An exception is an error condition during a program execution. PL/SQL supports programmers to catch such conditions using Exception block in a the program and an appropriate action is taken against the error condition. There are two types of exceptions.

USER-DEFINED EXCEPTIONS.

PLSQL allows you to define your own exception according to the need of your program. A user-defined exception must be declared and then raised explicitly, using either · RAISE Statement or the procedure DBMS-STANDARD.

RAISE_APPLICATION_ERROR.

The syntax for declaring an exception is

DECLARE

my-exception EXCEPTION;

Example :-

The following example illustrates the user-defined exception. This program asks for a customer ID, when the user enters an invalid ID, the exception invalid_id is raised.

DECLARE

```
C-id customers.id%type := &cc_id;
C-name customers.name%type;
C-address customers.address%type;
```

Q. What

is procedure? Explain different parameters in procedure with example.

A stored procedure is a group of SQL statement that form a logical unit and perform a particular task, and they are used to encapsulate a set of operations or query to execute a database server.

realme Shot on realme C33

Q-9. What is cursor? what are the different types of cursor.

When oracle process an SQL statement, it needs to allocate memory called context area (which is part of the program global area (PGA) allocated on the server). Cursor is a handle (or pointer) to the context area (PLSQL program can control the context area using cursor).

Two Types of Cursor:-

- 1) Explicit Cursors.
- 2) Implicit Cursors.

1) Explicit Cursors.

Explicit Cursors are programmed-defined cursors for gaining more control over the context area. An explicit cursor should be defined in the declaration section of the PLSQL Block.

- Declaring the cursor for initializing the memory.
- Opening the cursor for allocating the memory.
- Fetching the cursor for retrieving the data.
- Closing the cursor to release the allocated memory.

2) Implicit Cursor.

Implicit cursors are automatically created by Oracle whenever a SQL statement is executed. When there is no explicit cursor for a statement.

Whenever a DML statement (INSERT, UPDATE and DELETE) is issued an implicit cursor is associated with this statement. For, INSERT operations, the cursor holds the data that needs to be inserted. For UPDATE and DELETE operations, the cursor identifies the rows that would be affected.

realme

Shot on realme C33

Q-10. What is function? write the syntax for creating function and give an example.

A function is same as a procedure except that it return a value. The CREATE FUNCTION statement enables you to create a function in oracle. The function has a return clause that allows you to return a value to the calling program.

Syntax:-

A standalone function is created using the CREATE FUNCTION statement. The simplified form the CREATE OR REPLACE PROCEDURE Statement is as follows:-

```
CREATE (OR REPLACE) FUNCTION function_name  
[parameter_name [IN] | OUT | IN OUT] type[, ...]  
RETURN return_type  
[ISIAS)
```

```
BEGIN
```

```
<function_body>
```

```
END [function_name];
```

Example:-

The following example illustrates how to create and call a standalone function. This function return the total number of CUSTOMERS in the customers table.

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	delhi	1500.00
3	Kaushik	23	Kota	2000.00
4	Chaitali	25	mumbai	6500.00
5	Naradik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00

105

```
CREATE OR REPLACE FUNCTION total_customers
RETURNS number IS
total_number(2) := 0;
BEGIN
SELECT Count(*) into total
FROM customers;
RETURN total;
END;
```

realme Shot on realme C33

Q11 What is Triggers? what are different types of triggers?

The triggers can be used for many purpose such as imposed security on the use of database objects during a particular time, restricting user from deleting certain data from the database.

The events that fire a trigger include the following:

- DML statement that modify data in a table (INSERT, UPDATE, and DELETE)
- DDL Statement
- System events such as startup, shutdown, and error messages.
- User events such as logon and logoff.

BENEFITS OF TRIGGERS:

- Triggers can be written for the following purpose.
- Generating some derived column values automatically.
- Enforcing referential integrity
- Event logging and storing information on table access
- Auditing
- Synchronous replication of tables.
- Imposing security authorizations.
- preventing invalid transactions.

12. What is Aggregate function?

An aggregate function allows you to perform calculation on a set of values to return a single scalar value. We often use aggregate functions with the GROUP BY and HAVING clauses of the SELECT statement.

The following are the most commonly used SQL aggregate functions:

- AVG - calculates the Average of a set of values.
- COUNT - counts rows in a specified table or view.
- MIN - gets the minimum value in a set of values.
- MAX - gets the maximum value in a set of values.
- SUM - calculates the sum of values.

Note that all aggregate functions above ignore null values except for the COUNT function.

Syntax.

To call an aggregate function, use the following syntax:

- aggregate-function (DISTINCT | ALL expression)
- First, specify an aggregate function that you want to use. Then specify MIN, MAX, AVG, SUM or COUNT.
- Second, put DISTINCT or ALL modifier followed by an expression inside parentheses. If you explicitly use DISTINCT modifier, the aggregate ignores duplicate values and only consider the unique values. If you ALL modifier, the aggregate function uses all values for calculation or evaluation. The ALL modifier is used by default if you do not specify any modifier explicitly.

13.What is positivity level in function:

When you call a function using the SQL statement , Oracle checks if the called function is executable. A function is executable when the modification made to the database objects or variables to which the function refers exists .The side effects increase the time to execute a query or function when you declare user session packaged variables.

Oracle applies certain restrictions when you call when you call stored functions from SQL Statement , such as:

- A function cannot modify database tables without executing DML Statement , as as INSERT , UPDATE , or DELETE.
- You cannot execute a function remotely or in parallel that reads or writes the value of packaged variables.
- Functions called from select .values or a set clause is entitled to modify the value of packaged variables.
- A function can not call another sub program or can not reference a view that does not imply with the restrictions.

Q. List and Explain all sections of PL/SQL block?

PL/SQL blocks can be nested within other PL/SQL block using BEGIN and END. Every PL/SQL statement ends with a semicolon (;) .

Basic structure of a plsql block -

```
DECLARE  
    Executable declaration section  
BEGIN  
    Executable command(s)  
    Exception handling  
END;
```

Declaration :- This section starts with the keyword DECLARE. It is an optional section and defines all variable, cursor, subprograms , and other elements to be used in the program.

Executable Commands : This section starts with the keyword EXECUTEABLE commands in the program. PL/SQL statements of the program. It consists should have at least one executable line of code . which may be just a null command to indicate that nothing should be executed.

Exception Handling : - This section starts with the keyword handle errors in the programs.

- i) END keyword that ends the PLSQL block .A semi colon
- ii) the END marking the end of the block.

15. Explain CODD's rule?

Rules:-

Rule 1 :- The Rule of Relational Database Management Rule

- Any Relational Database Management System must use only its relational capabilities to manage the information stored in the database.

Rule 2 :- Rule of Information

- All information in Relational Database must be represented as values within columns and rows .Tables are relational Database management systems. If any user defined data as meta - data , it is important to store the value as an entry in the table cells.

Rule 3 :- Rule of Guaranteed Access.

- All data in a database should be logically accessible by using a combination of table name , column name and primary key for the table.

Rule 4 :- Rule of Systematic Null Value Support

- Null values are completely supported in relational databases. They should be uniformly considered as 'missing information'. Null values are independent of any data type. They should not be mistaken for blanks or zeroes or empty strings.

Rule 5 :- Rule of Active and Online Relational Catalog

- The representation of database structure as a collection of tables as catalog or data dictionary. The active online catalog that stores the active meta- metadata is called data dictionary .These data dictionary .These data dictionary is accessible only by authorized users who have the required privileges and the query language used for accessing the database should be used for accessing the data dictionary.

Shot on realme C33

Rule 5: Rule of Comprehensive data sub language.
It should support data manipulation and definition, integrity constraints, transaction processing and authorization. All functional database terms of standard structured query language (SQL).

Rule 6: Rule of updating views.
Views should reflect the updates of their respective base tables and vice versa. A view is a logical table which shows restricted data. Views make the data readable but not modifiable. Views help in closer abstraction.

Rule 7: Rule of Self Level Insertion, Update and deletion.
A single operation should be sufficient to receive, insert, update or delete the data. The insert, update and delete operations can be supported for any executable set rather than just single row in single table.

Rule 8: Rule of Physical order Independence.
The database user need not know about the physical implementation details storage and retrieval. Any to the physical structure of the database should not effect the database user's method of accessing the data. Batch and end users operations are logically separated from storage and respective access methods.

Rule 9: Rule of Logical Data Independence.
The database user's view of data affected by any change in the base tables. This is one of the most difficult conditions to achieve since database rely on strong ties between database users and the logical view of the data.

Rule 10: Rule of Integrity Independence.
Data Integrity rule must be defined in the relational database and should be available and stored in data dictionary and not in the application programs.

Shot on realme C33

Rule 11: Rule of Distribution Independence.
Distributed database enable you to stored data across multiple databases. These database can be either available on the local machine or some remote machine. The data manipulation language of the relational system should not be connected about the physical data storage and no alterations should be required if the physical data is centralized or disturbed.

Rule 12: Rule of Non Subversion.
Any row should obey the security and integrity constraints imposed. No special privileges are applicable.

realme Shot on realme C33

Q6. Explain CASE statement with example in Oracle

The CASE statement choose from a sequence condition , and executes a corresponding statement. the CASE Statement evaluates a single expression and compares it against several potential values , an evaluate multiple Boolean expression and chooses the first one that is TRUE.

Syntax :- Simple - CASE- statement :-

If the Else clause is omitted , the system substitutes a default action . for a CASE a statement , the default when none of conditions matches is to execute a CASE - NOT - FOUND exception . for CASE expression , the default is to return null .

Example : The following example shows a searched CASE statement realme . The WHEN clauses can use different conditions and then all testing the same variable or using the ELSE operator . Be aware this example does not use an ELSE clause , an exception is raised if none of the WHEN conditions true next .

Shot on realme C33

```
DECLARE
    quantity NUMBER := 100;
    projected NUMBER := 50;
    needed NUMBER := 999;
```

BEGIN

<< here >>

```
CASE
    WHEN quantity IS NULL THEN
```

```
aboms - output .put - line (' quantity not available ' );  
when quantity + projected > = needed THEN  
aboms - output .put - line (' quantity ' || quantity ||  
' should be enough if projections are met ' );
```

```
when quantity >= 0 THEN  
aboms - output .put - line (' quantity ' || quantity ||  
' is probably not enough ' );
```

```
END CASE here;
```

Exception .

```
ENI CASE -NOT -FOUND THEN  
aboms - output .put - line (' somehow quantity is the  
than 0: ' );  
END;
```

realme | Shot on realme C33

17) Explain different date function with example.

Format Code	Description	Range of values
DD	Day of the month	1 - 31
DY	Name of the day in 3 uppercase letters	SUN..., SAT
DAY	Complete name of the day in uppercase, padded to 9 characters	SUNDAY..., SATURDAY
MM	Number of the month	1-12
MON	NAME of the month in 3 uppercase letters	JAN,..., DEC
MONTH	Name of the month in uppercase padded to a length of 9 characters.	JANURY..., DECEMBER
RM	ROMAN numeral for the month	I..., XII
YY or YYYY	Two or four digit year	71 or 1971
HH:MI:SS	Hours : Minutes : Seconds	10:28:58
HH 12 or HH 24	Hour displayed in 12 or 24 hour format	1-12 or 1-24
MI	Minutes of the hour	0-59
SS	Seconds of the minute	0-59
AM or PM	Meridien indicator	AM or PM
SP	A suffix that forces number to be spelled out. A suffix meaning that the cardinal number is to be added	e.g. TWO THOUSAND NINE e.g. 1st, 2nd, 3rd,...
FM	Prefix to DAY or MONTH or YEAR to suppress padding	e.g. MONDAY with no extra spaces at the end.

Example: Select TO_CHAR(SYSDATE, 'MON DD,YYYY') FROM DUAL;

Select TO_CHAR(SYSDATE, 'HH24:MI:SS AM') FROM DUAL;

Select TO_CHAR(NEW_TIME(SYSDATE, '+01:00'), 'HH24:MI') FROM DUAL;

TO_DATE ('SEP 27, 1999', 'MON DD, YYYY'),

TO_DATE ('13-MAR-2009', 'DD-MON-YYYY')

FROM DUAL;

Select NEXT_DAY(SYSDATE, 'FRIDAY') FROM DUAL;

Select LAST_DAY(ADD_MONTHS(SYSDATE, 1)) FROM DUAL;

Q Explain different string functions with example.

Function	Input Argument	Value Returned
INITCAP (s)	s = character string	First letter of each is changed to uppercase and all other letters are in lower case.
LOWER (s)	s = character string	All letters are changed to lowercase
UPPER (s)	s = character string	All letters are changed to uppercase
CONCAT (s1, s2)	s1 and s2 are character strings	Concatenation of s1 and s2 Equivalent to s1 s2
LPAD (s1, n [s2])	s1 and s2 are character strings and n is an integer value	Returns s1 right justified and padded left with n characters from s2, s2 defaults to space.
RPAD (s1, n [s2])	s1 and s2 are character strings and n is an integer value	Returns s1 left justified and padded right with n characters from s2, s2 defaults to space.
LTRIM (s [set])	s is a character string and set is a set of characters	Returns s with characters removed up to the first character not in set ; defaults to space.
RTRIM (s [set])	s is a character string and set is a set of characters	Returns s with first characters removed after the last characters not in set ; defaults to space.
REPLACE (s, search_s [, replace_s])	s = character string , search_s = target string , replace_s = replacement string.	Returns s with every occurrence of search_s replaced by replace_s ; default removes search_s
SUBSTR (s, m [, n])	s = character string , m = beginning position n = number of characters	Returns a substring from s beginning in position m and n characters long ; default returns to end of s.

LENGTH (S)	S = character string	Returns the number of character in S.
INSTR (S1, S2[, m[, n]])	S1 and S2 are character strings, m = beginning position, n = occurrence of S2 in S1	Returns the position of the nth occurrence of S2 in S1 beginning at position m, both m and n default to 1.

Example:- select concat ('Alan', 'Turing') as 'NAME' from dual;
 select 'ALAN || 'Turing' as "NAME" from dual;
 select initcap ("now is the time for all good men
 to come to the aid of the party") as "SLOGAN"
 from dual;
 Select substr ('ALAN Turing', 1, 4) as first "from dual";

Q9 Explain different numeric functions with example.

20. Write difference between procedure and function.

Procedure

- i) Procedure are basic PL/SQL blocks to perform a specific action
- 2) Procedure will not return the value.
- 3) Procedure always executes as PL/SQL statement.
- 4) It does not contain return clause in headers section.
- 5) we can pass the values using IN, OUT, IN OUT parameters
- 6) procedures can not be executed in select statement

Function

- i) functions are blocks used mainly to perform the computation
- 2) Functions must return the value. when you are writing functions make sure that you can write the return statement.
- 3) Functions executes as part of expression.
- 4) It must contain return clause in headers.
- 5) functions must return a single value.
- 6) FUNCTION can execute or call using select statement but it must not contain IN or OUT or IN OUT procedures.