**STRUCTURED QUERY LANGUAGE**

**Introduction**

**Categories:**

* Data Definition Language.
* Data Manipulation Language.
* Transaction Control Language.
* Data Control Language.
* Data Retrieval Language.

**Data Definition Language (DDL):** To define the database objects in Database.

* Create
* Alter
* Drop
* Truncate
* Rename

**Data Manipulation Language (DML):** To manipulate the data in the database. (to play with the data).

* Insert
* Update
* Delete

**Transaction Control Language (TCL):** To control the transactions or query executed.

* Commit 🡪 saving the transactions.
* Rollback 🡪 Undoing the transactions.
* Save point. 🡪 Capturing the transactions.

**Data Control Language (DCL):** To control the access levels of data. (to see or access the data)

* Grant
* Revoke

**Data Retrieval Language. (DRL):** To retrieve the data from database.

* Select 🡪 To view the data.

**ORACLE DATABASE (RDBMS)**

1. **DB Mode**
2. **DB Logical View**
3. **DB objects**

**DB Mode:**

* Open (or) Read/Write Mode: Users can use the DB.
* Mount Mode: DBA can use the DB for maintenance / Upgradation/migration. ( for Administration, DBA teams ) which is build mode.
* Inactive Mode: None can use the DB; this is where prerequisites are defined.

E.g.: Sand 🡪 Clay pottery 🡪 POT (inactive mode 🡪 Mount mode 🡪 Open mode)

**DB logical View:**

Database 🡪 tablespace 🡪 segment🡪 block ref 9.10

Eg: Apartments(server)🡪 floors (tablespaces)🡪houses(segment)🡪 rooms(block)

**Database objects list:**

1. Tables
2. Views
3. Functions
4. Indexes
5. Sequences
6. Materialized views
7. Triggers
8. Stored procedure
9. Packages

**Tables:**

It consists of rows and columns

**Views:**

To view selected or particular rows and columns from multiple tables, if u want to run a query daily you can save this view query in database itself. View is a query which can refer one or more tables, one or more columns from multiple tables and give u result like a tabulation.

**Functions:**

Smallest logical pockets

**Indexes:**

To retrieve the data from huge database as quick as possible. (fast retrieval of data from database)

**Sequences:**

Generate the series of numbers.

**Materialized views:**

You write a query; you get end result, and you store that result in database itself. The result of the view or query is stored is called Materialized views. (Query stored along with the result).

**Triggers:**

Action based on an event(e.g.: alarm clock)

**Stored Procedures:**

Kind of storage, you store some query and name it as some variable. (eg: Lots of food idly sambar, dhal, papad, pickle here you separate the foods and name it as north Indian and south Indian foods, so here north Indian and south Indian are stored procedure). It is elaborated procedures.

**Packages:**

It will have series or sequence of stored procedure inside a package.

**DATA:**

**Types:**

1. Char
2. Varchar
3. Integer
4. Decimal
5. Date
6. Timestamp
7. Clob
8. Blob
9. Bfile

**Char/Varchar**: char allows only the alphabets, where Varchar allows alphabets,number, symbol.

Char(3) max length is 3.

**Number and decimal**: number : 173226

Decimal: [decimal(p,s)] eg : 173226.62 – decimal ( 8,2)

P – precision is total count of the number, s – scale is total count after the decimal.

**Date and Timestamp:**

**Date –** date is only date. – DD- MON- YYYY

**Timestamp:** date and time – DD – MON -YYYY HH:MM:SS:SSSSSS

**CLOB:** Character Large objects e.g.: resume

**BLOB:** Binary large Objects e.g.: Photo

**BFILE:** Binary file e.g.: Audio or Video file

**Constraints:**

1. Primary key
2. Foreign key
3. Unique key
4. Not null
5. Check

**Primary Key:**

* Only **one** column can be declared as primary key in a table.
* No duplicates are allowed.
* No nulls are allowed.
* Auto indexed. ( indexing fast retrieval of data) automatic indexing.

**Unique Key:**

* **More than one** column can be declared as unique key in a table.
* No duplicates value is allowed.
* Null values are allowed.
* Auto indexed.

**Foreign Key:**

* **More than one** column can be declared as foreign key in a table.
* Duplicates allowed as they are referenced values.
* Null values are allowed as they are referenced values. Ref 3.40

Eg: if I am traveling to some other country then I am a foreigner in their point of view. If some other country people visits India then they call it as foreigner.

So, the data which is present inside the primary key column in this table is called primary key/ value/data. And when it is referenced to some other table then it’s a foreign key.

**NOT NULL:**

* More than one column can be declared as not null column in a table.
* Duplicate values are allowed.
* No null values are allowed.

**Check**:

* Any column can be checked for its value. E.g. salary > 20000
* This allows only the values greater than 20000 in the declared column.

A screen shot of a computer

Description automatically generated with medium confidence

**SQL:**

1. Create – creating the database.
2. Alter – created a table and you want to adding column or records(rows).
3. Drop – dropping the whole table.
4. Truncate – you want only the structure of the table not the records.
5. Rename – renaming the database.
6. Insert – enter the data into the table.
7. Update – already you have entered the records in columns, and you want to change the record of the column.
8. Delete – to delete the selective record of the table.
9. Commit - saving the transactions. (Transactions – add or delete the data)
10. Rollback - Undoing the transactions.
11. Save point – saving from particular point, capturing and saving the transactions.
12. Grant – granting the access (giving the access)
13. Revoke – removing the access (take the access)
14. Select – to view the data.

**Data Definition Language:**

Create:

Type 1:

create table #emp (id int identity(1,1), name varchar(3), age int,emp\_id int)

select \* from #emp

Type 2:

create table #employee (id int identity(1,1), name varchar(3) not null, age int,emp\_id int primary key)

constraints is used to validate whether it is a valid data. Data type is used to put the right kind of data. Ref 13.05.

**Type 3:**

create table #employee (id int identity(1,1), name varchar(3) not null, age int check (age >18),

emp\_id int primary key, company\_name varchar(10) References #emp (company\_name))

select \* from #employee

if you name constraints it would more easy to perform on later purpose. It is better to name the constraints.

A computer screen with blue and white text

Description automatically generated with low confidence

Type 5:

create table #empeen as select \* from #employee

for backup purpose.

**Alter Statements:**

A picture containing text, screenshot, document

Description automatically generated

**Cascade:** deletes the primary key along with foreign key.

**Disable:** Temporarily disable the connection between the primary and foreign key.

**Enable:** Connects between primary key and foreign key.

A screenshot of a computer

Description automatically generated with medium confidence

**DATA MANIPULATION LANGUAGE**: To manipulate the data in the database.

1. Insert
2. Update
3. Delete

Insert :

Using & symbol before the column name will prompt us (pop up will appear to enter the values) ref 9.26

A picture containing text, screenshot, display

Description automatically generated

**Update:**

Create table #emp( id int identity(1,1), Name varchar(14),age int,portal int, dep varchar(10))

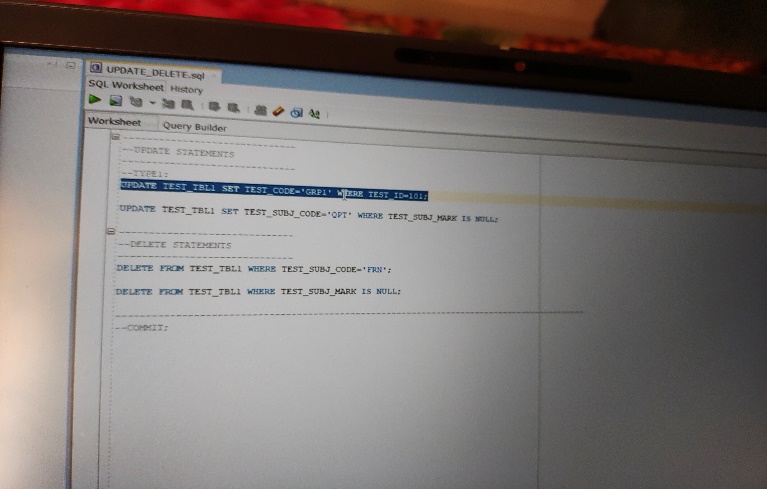
insert into #emp (Name,age,portal,dep) values('priya',20,180100,'IT'),('tamil',22,180364,'IT'),('deepu',18,202446,'IT')

select \*from #emp

update #emp set [portal]=202556 where [portal]=202446

**Delete:**

delete from #emp where [portal]=202556



**Transaction Control Language:** Commit (Save), Rollback(Undo), Save point(save certain set of actions).

A picture containing text, electronics, display, screenshot

Description automatically generated

Commit will save your commands permanently.

**Data Control Language: To control the access level of data.**

Grant, Revoke. (Administrative Tasks)

A picture containing text, computer, display, screenshot

Description automatically generated

**Data Retrieval Language:** To retrieve the data from database.

A picture containing text, electronics, display, computer

Description automatically generated

**Functions:**

* SQL functions are built into Oracle Database and are available for use in appropriate SQL statements.
* **NOTE:** SQL functions are different from the user defined functions in PLSQL.

**TYPES OF FUNCTIONS:**

1. String/char functions
2. Numeric/Math functions
3. Date/Time functions
4. Conversion functions

String/Char functions:

* CONCAT: To add 2 strings.
* INITCAP: to capitalize only the first letter in the string.
* INSTR: to find the position of string in a string.
* SUBSTR: to extract part of the string.
* LENGTH: to find the length of the string
* LOWER: to convert the string into lower case
* UPPER: to convert the string into lower case
* TRIM: to trim the string.

A screenshot of a computer

Description automatically generated with medium confidenceA picture containing text, screenshot, menu, fabric

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

NUMERIC/MATH Functions:

1. COUNT: To count the number of records
2. ROWNUM: To get the row number of the selected result or table.
3. SUM: to sum up the values of any selected column
4. TRUNC: To truncate/ trim the numeric value.

*2 Image captured.*

DATE /TIME FUNCTIONS:

1. SYSDATE: To display system date.
2. SYSTIMESTAMP: To display system date and time.
3. CURRENT\_DATE: To display current date.
4. CURRENT\_TIMESTAMP: To display current date and time.
5. LAST\_DAY: To display last date of the month or any given date value.

1 Image captured:

CONVERSION FUNCTIONS:

1. TO\_DATE
2. TO\_CHAR
3. TO\_NUMBER

1 image captured:

JOINS:

Joins are used to retrieve data from multiple tables. It is performed whenever tow or more tables are joined in SQL statement. ( if you want to select data from multiple tables you can use joins.

**TYPES:**

1. Oracle inner join (sometimes called simple join)
2. Oracle left outer join (sometimes called Left join)
3. Oracle right outer join (sometimes called right join)
4. Oracle full outer join (sometimes called full join)

Graphical representation:

1 image captured

Join statements:

1 image captured

**Introduction To DBMS:**

* **Data:** Known fact, recorded with implicit meaning (Processed information)
* **Database:** Collection of logically related data. ( collection of logically related data)
* **Database Management System:** A computerized record- keeping system. ( to manage database)

**Note:**

* Computer based information system (IS) can serve a variety of complex tasks in a coordinated manner.
* Such system handles large volumes of data, multiple users and several applications for activities occurring in a central and /or distributed environment.
* DBMS provides for storage, retrieval and updation of data in an organized manner.

**Goals:**

* To provide an environment which will efficiently provide access to data in a database.
* Enforce information security – database security, concurrency control, cash recovery.

**General purpose:**

* Defining database – specifying the types of data to be stored in the database.
* Constructing database - storing the data
* Manipulating database – querying the DB, updating the DB and generating the reports from DB.

**Why database?**

* File management system had drawbacks which lead to database.

**Drawbacks:**

* Data is separated for every individual program.
* Data can be duplicated as these are for individual program.
* No uniformity can be found in the data.
* All the data would have been arranged as per the need of the specific program.
* No security control.

**Benefits of DBMS:**

* Redundancy is reduced.
* Inconsistency is avoided.
* Data is shared.
* Standard is enforced.
* Security is applied.
* Integrity is maintained.
* Data independency is provided.

**DATABASE MANAGEMNET SYSTEM:**

* An agent that allows communication of the users with the physical database and the operating system (without the users knowing how it is done)
* Metadata is data about the DATA stored in DB, i.e., structure of data stored in the DB.

Users 🡪 application software/queries 🡪 processing queries 🡪 software to access storage data🡪 1. metadata, 2. Database.

**3 broad classes of users:**

**Application programmers:**

* Programmers write application to use the database.

**End users:**

* Users who query the DB using the database applications

**Database administrators (DBA):**

* Administrators create the actual DB and maintain the DB (like performance, technical, services need to enforce the policy decisions)

**What happened when we query DB?**

User query 🡪 query processors 🡪 database manager 🡪 file manager 🡪 Physical DB

* Query processors – translate queries to low- level instructions.
* Database manager – interface between queries and the low- level data stored.
* File manager – represents information stored on the disk (allocation of space and data structures)

**3 TYPES OF DATABASE SYSTEMS: DATA MODELS**

**Data model:** The main tool which provides the data abstraction (database approach) by hiding the details of data storage which is not needed by the users i.e., set of concepts used to describe the structure of the database.

**Categories:** As per data structure.

**High level/ conceptual** – provides details on the way users perceive the data. Using entity, attribute and relationship.

**Low level/ Physical** – provides details on how data is stored. Indexing which makes the search efficient based on the structure.

**Representational / implementation** - Between above 2 extremes, that provides details on both which are understood by end user and not too many details are hidden.

**DATABASE ARCHITECTURE:**

There are 3 levels of architecture:

**External** – the one at the user end which is connected how the user views the data.

**Internal** - the one at the Physical DB end which is concerned on how the data is stored in DB.

**Conceptual** – the one between the above mentioned which is concerned on how they are connected.

DB architecture:

1 image captured:

**SCHEMA:**

Descriptions of data in terms of Data model.

**3 level DB architecture:**

* External schema
* Conceptual schema
* Internal schema

Types of Database models:

The most known record- based models are Hierarchical, Network and relational models.

Major types:

Hierarchical – information management system from IBM.

Network – integrated data store

Relational – Oracle, Sybase, MY SQL etc.,

Object oriented – versant object database, Object store and ZODB.

1 image captured ( 0.36)

**RELATIONAL DATABASE MANAGEMENT SYSTEM: (Relational Mode)**

Relational database management system (Relational model)

It is a system in which:   
-The data is perceived by the user as tables (as nothing but tables)

-The operators at the user’s disposal – e.g. for data retrieval – are operators that generate new tables from old, and those include at least SELECT, PROJECT AND JOIN.

**CODD RULES:**

1. Information rule
2. Guaranteed access rule
3. Systematic treatment on null values
4. Dynamic on-line catalog based on the relational model.
5. Comprehensive sub language
6. View updating rule.
7. High level insert update and delete rule.
8. Physical data independence rule
9. Logical data independence rule
10. Integrity independency rule
11. Distribution rule
12. Non subversion rule

**FEATURES:**

* The ability to create multiple relations (tables) and enter data into them.
* An interactive query language
* Retrieval of information stored in more than one table.
* Provides a catalog or dictionary, which itself consists of tables (called system tables)

**IMPORTANT TERMS:**

* Relation: a table
* Tuple: a row in a table
* Attribute: a column in a table
* Degree: number of attributes
* Cardinality: number of tuples
* Primary key: a unique identifier for the table
* Domain: a set of possible values that a given attribute can have

KEYS:

* Simple key vs composite key – single attribute vs combination of attributes
* Key – an attribute or set of attributes whose values uniquely identify each entity
* Super key- adding additional to a key , an augmented key is super key
* Candidate key – combination of attributes

Example of super key:

1 image captured : ref 6.15 in rdbms

**ENTITY RELATIONSHIP MODEL:**

**Entity –** anything that exists and distinguishable. (anything that existing addresses inside the database is entity ) eg : tables are entity

**Entity set** – a group of similar entities.

**Attribute** – properties that describes entity. ( properties , which is inside the column)

**Relationship** – an association between the entities

Notations:

Entity Type Weak entity type Attribute type Relationship type

Weak relationship type

Entity model : ref module 2 : 23.54 1 image captured

Relationship model : ref module 22.49 1 image captured