**SQL:**

* Data & Database:
  + Hierarchical, Network, Relational
* DESC, INFO commands
* NULL value means absence of value, or value not known, cannot be compared to any other value, not even another NULL value
* As keyword : Alias, column and Table both can be aliased, generally ‘as’ is used for column alias
* ||: Implicit concatenation Pipes
* DISTINCT
* WHERE
* <> & !=
* BETWEEN … AND
* IN, NOT IN
* LIKE: Inexact search, % and \_ : single char & 0 or more chars
* IS NULL
* AND, OR
* ORDER BY multiple cols, ASC and DESC
* Row functions: Case manipulation & Character manipulation:
  + LOWER, UPPER, INITCAP, CONCAT, SUBSTR
  + LENGTH, INSTR, LPAD, RPAD, TRIM, LTRIM, REPLACE
* ROUND, TRUNC, CEIL, FLOOR, First two functions work on numbers and date types.
* MONTHS\_BETWEEN, ADD\_MONTHS, NEXT\_DAY, LAST\_DAY
* SYSDATE
* TO\_CHAR, TO\_NUMBER, TO\_DATE
* NVL, NVL2, NULLIF, COALESCE
* DECODE
* Aggregation functions
* GROUP BY
* HAVING
* WINDOW functions: Window functions can create a window (over a group of similar values in a column, just like the GROUP BY clause), over which we can do some calculations. The most common of these calculations are Running Sum, RANK, LEAD, LAG, NTILE,COUNT.
* CASE statement: This statement creates a new column for our output. CASE WHEN some\_condition THEN something ELSE something\_else END.
* COUNT and CASE are used in conjunction
* JOINS: Equijoin, LEFT, RIGHT, OUTER, USING clause
* DML: Data manipulation language, structure of the database is not changed, the data in the database is altered, **row level changes are most common**:
  + **Insertion** of a Row: INSERT INTO table(column\_names) VALUES(vals…); Column names are optional, the values must align with the data types of the Columns
  + **Updating**: UPDATE table SET column\_2=”new\_value” WHERE column\_1=”some\_value”: If the condition in the WHERE clause is not specified, then all the rows will have an updated column\_2 value of “new\_value”
  + **Deletion**: Delete Rows which satisfy a particular condition. DELETE FROM table WHERE column\_2=”some\_value”. If the WHERE condition is not mentioned, then all the Rows in the table will be deleted.
  + **Merge**: We are merging from one table into another. If the particular row already exists, then Merge follows an updating procedure, else it adds a new row.
* Transaction Control: A database transaction starts when DML statement is executed. It ends when:
  + **COMMIT** or **ROLLBACK** is issued
  + A **DDL/DCL** statement runs
  + **System Failure**
  + **COMMIT**: To ensure DML changes to the database COMMIT is necessary. COMMIT;
  + **ROLLBACK**: Rollback any changes. DELETE FROM table WHERE col=”some\_value”; ROLLBACK; Then the deletion operation is rolled back
  + **SAVEPOINT**: After the Query: SAVEPOINT s1; Intermediate steps in the Changes procedure. Once we have committed a savepoint, we cannot roll it back. **ROLLBACK TO SAVEPOINT s1**; COMMIT;
* DDL: Data definition Language, modify the structure of the database. CREATE, ALTER, DROP.
  + **CREATE** TABLE tablename (col\_name col\_type); A Table can also be created from another Table
  + **DROP** TABLE table\_name; Data as well as structure of the Table is deleted
  + **SELECT** table\_name FROM user\_tables ; Names of the Tables which the user owns
  + **ALTER**: **Add** a new column, **modify** data type of a column, or **Drop** the Column completely
  + **RENAME** table TO new\_name
  + **TRUNCATE** TABLE table; Same as deleting without a WHERE clause.
* **Constraints**: Limit the type of data which can go into a Table. During CREATE or after table creation using ALTER, we can specify the constraints.
  + CREATE TABLE table\_name (col col\_type, CONSTRAINT col PRIMARY KEY(col)); **Primary Key** constraint added to the col of the Table table\_name
  + CONSTRAINT col FOREIGN KEY(col) REFERENCES other\_table(other\_col) : col column of the current table is under a **referential integrity constraint**, deriving it’s values from other\_col from other\_table.
  + NOT NULL constraint : Only column level and not table level. Col col\_type **NOT NULL**
  + Example of NOT NULL constraint: CREATE TABLE emp (hiredate DATE NOT NULL)
  + **CHECK** constraint to make sure that a column contains values lying in a specified range.
  + **ALTER** command: ADD, DROP DISABLE, ENABLE constraint
* UNION, UNION ALL, MINUS, INTERSECT SET Operators
* At first glance, this may sound like a view, but views and temporary tables are rather different: A view exists only for a single query. Each time you use the name of a view, its table is recreated from existing data. A temporary table exists for the entire database session in which it was created.
* CREATE VIEW view\_name AS SELECT …
* Data can be retrieved from a View.
* DROP VIEW view\_name
* Sequence is a set of integers 1, 2, 3, … that are generated and supported by some database systems to produce unique values on demand.
* A sequence is a user defined schema bound object that generates a sequence of numeric values.
* Sequences are frequently used in many databases because many applications require each row in a table to contain a unique value and sequences provides an easy way to generate them.
* The sequence of numeric values is generated in an a**scending or descending order** at defined intervals and can be configured to restart when exceeds max\_value.
* INDEX is an ordered list for fast searching, a unique constraint or a primary key constraint automatically creates an INDEX.
* **User Access**:
  + **System Privileges: Creating, removing users, removing Tables**. Data base admin can assign these privileges.
  + CREATE USER scott IDENTIFIED BY scott\_password;
  + **GRANT** … TO scott; … may have **CREATE SESSION, CREATE TABLE, CREATE SEQUENCE, CREATE VIEW, CREATE PROCEDURE** privileges.
  + **REVOKE** … FROM scott; … may have CREATE SESSION, CREATE TABLE, CREATE SEQUENCE, CREATE VIEW, CREATE PROCEDURE privileges.
  + **Object Privileges**: Assign the right to perform a particular operation on a specific object. **SELECT, INSERT, UPDATE, DELETE.**
  + GRANT SELECT ON employees TO scott,amy;
  + GRANT SELECT ON alice.departments TO PUBLIC; All users on the systems to query data from Alice’s DEPARTMENTS table.
  + Similar to GRANT, REVOKE will revoke privileges from a particular user.
  + **Roles**: A user can have access to several roles and the same role can be accessed by several users. Several privileges can be granted to or revoked from the user.
  + CREATE ROLE manager;
  + GRANT … TO MANAGER
  + GRANT/REVOKE role TO/FROM scott;
  + **ALTER USER scott IDENTIFIED BY lion; Changing Scott’s password**
* Sub Queries, Query inside another Query. Single Row and Multiple Row subqueries. Correlated Subqueries
* ALL & ANY used with inequality operators. IN is by default used with the = EQUALITY operator.
* EXISTS operator: Returns TRUE if the argument to the EXISTS operator contains atleast one Row. EXISTS(SELECT \* FROM emp WHERE mgr=outer.emp\_no)