SEQUENTIAL QUERY LANGUAGE CHEAT SHEET

SQL Basics

SQL

Structured query language (SQL) is a domain specific language used for programming and querying a database

SQL Data Types

Exact Numeric's:
- INTEGER
- SMALLINT
- BIGINT
- NUMERIC
- DECIMAL

Approximate Numeric's:
- REAL
- DOUBLE PRECISION
- FLOAT
- DECFLOAT

Binary Strings:
- BINARY
- BINARY VARYING
- BINARY LARGE OBJECT

Boolean:

Intervals:
- INTERVAL DAY
- INTERVAL YEAR

Character Strings:
- CHARACTER
- CHARACTER VARYING (VARCHAR)
- CHARACTER LARGE OBJECT
- NATIONAL CHARACTER
- NATIONAL CHARACTER VARYING
- NATIONAL CHARACTER LARGE OBJECT

Date times:
- DATE
- TIME WITHOUT TIMEZONE
- TIMESTAMP WITHOUT TIMEZONE
- TIME WITH TIMEZONE
- TIMESTAMP WITH TIMEZONE

Collection Types:
- ARRAY
- MULTISET

Other Types:
- ROW
- XML

View

It is a virtual table which is a result of a query. It is often used as a security mechanism letting users to access the data through the views

Syntax:

```
CREATE VIEW view1 AS
SELECT c1,c2
FROM t1
WHERE condition
```

Function	Description
TO_DATE	It is used to convert a string to date.
COALESCE	Returns the first non NULL results, when querying with the columns that contain NULL
CURRENT_TIMESTAMP	Returns the correct time on the database server
COUNT	An aggregate function that returns the number of rows in the result set
SUM	An aggregate function that sums up the values in a result set
AVG	To compute the mean average of the values in the result set
MIN/MAX	An aggregate function to return the largest/smallest value among the result set
LISTAGG	It is used to transform values from a group of rows into a delimited string

Functions

Aggregate Functions: It is a function where the values of multiple rows are combined to form a single value

UNION: A set operation can be used on the returned results called 'UNION' which can append the result of one query to another

Syntax:

```
SELECT col1, col2 FROM table1
UNION
SELECT col3, col4 FROM tabl2
```

INDEXES

It is used to speed up the performance of the queries by reducing the number of database pages to be visited

Syntax:

To create an index: `CREATE INDEX index_name ON t(c1, c2)`

To create an unique Index: `CREATE UNIQUE INDEX index_name ON t(c3, c4)`

To drop an index: `DROP INDEX index_name`

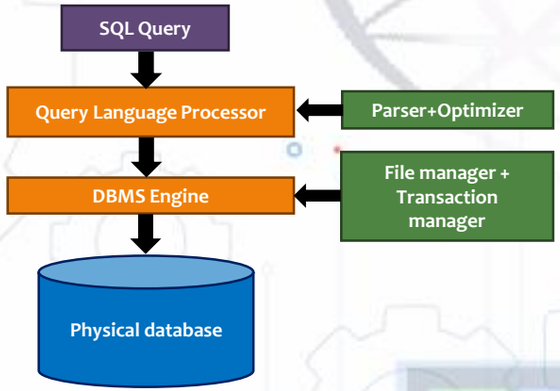
Stored Procedure

It is a set of **SQL** statements with assigned names that can be shared and reused by multiple programs

Syntax: To create Procedure

```
CREATE PROCEDURE procedure_name
 @variable AS datatype = value
 AS
 -- Comments
 SELECT * FROM t GO
```

Keywords	Explanation
SELECT	It is used to specify which column to query. Use * for all
FROM	It is used to declare the table to select from
WHERE	It is used to define a condition
=	Used to compare a value with the given input
LIKE	It is a special operator used with WHERE to search for a specific pattern from a column or row
GROUP BY	It is used to group identical data
HAVING	It is used to specify that rows with aggregate values which meets the specifies condition must be returned
INNER JOIN	It is used to return all rows where key records of one table is same as that of the other table
LEFT JOIN	It is used to return all rows from the left table with the matching rows in the right table
RIGHT JOIN	It is used to return all rows from right table with the matching rows in the left table
FULL OUTER JOIN	It is used to return rows that match either in the left or right table



Using SQL constraints

Primary Key: Set c1 and c2 as primary key

Syntax:

```
CREATE TABLE t(
    c1 INT, c2 INT, c3 VARCHAR,
    PRIMARY KEY (c1,c2)
    );
```

Foreign Key: Set c2 column as a foreign key

Syntax:

```
CREATE TABLE t1(
    c1 INT PRIMARY KEY,
    c2 INT,
    FOREIGN KEY (c2) REFERENCES t2(c2)
    );
```

Operator	Syntax	Description
UNION	SELECT C1 FROM t1 UNION [ALL] SELECT C1 FROM t2	Selecting column 1 for table t1 and column 2 from table t2 and combine the rows of these two queries
INTERSECT	SELECT C1 FROM t1 INTERSECT SELECT C1 FROM t2	It is used to return the intersection of two queries
MINUS	SELECT C1 FROM t1 MINUS SELECT C1 FROM t2	It is used to subtract the second result set from the first
NOT LIKE	SELECT c1 FROM t WHERE c1 [NOT] LIKE pattern	It is used to return the query of rows using the matching pattern
BETWEEN	SELECT c1 FROM t WHERE c1 BETWEEN min AND max	It returns the rows where c1 is between MIN and MAX
NOT NULL	SELECT c1 FROM t WHERE c1 IS [NOT] NULL	To check if the values are NULL or NOT NULL

Trigger

It is a special type of stored procedure that automatically executes when a user tries to modify through a DML event

Syntax:
- To create or modify trigger

```
CREATE OR MODIFY TRIGGER
trigger_name
WHEN EVENT
ON table_name TRIGGER_TYPE
EXECUTE stored_procedure
```

Explanation:
- WHEN:
  - BEFORE: Invokes before an event occurs
  - AFTER: Invokes after an event occurs
- EVENT:
  - INSERT: Invoke for Insert
  - UPDATE: Invoke for Update
  - DELETE: Invoke for Delete
- TRIGGER TYPE:
  - FOR EACH ROW
  - FOR EACH STATEMENT

- To delete or drop a trigger: Used to delete a specific trigger

Syntax: `DROP TRIGGER trigger_name`

Unique

Making the values in C1 and C2 as unique

Syntax:

```
CREATE TABLE t(
    c1 INT, c1 INT,
    UNIQUE (c2,c3)
    );
```