SQL & NOSQL CHEAT SHEET

SQL Cheat Sheet

Querying Data

- SELECT – Retrieve data from one or more tables
- DISTINCT – Select distinct values in a column
- WHERE – Filter rows using a condition
- AND, OR, NOT – Combine multiple conditions
- ORDER BY – Sort the result set by a column
- LIMIT & OFFSET – Limit the number of rows and skip rows
- GROUP BY – Group rows with the same values in specified columns
- HAVING – Filter the results of a GROUP BY

Joining Tables

- INNER JOIN – Combine rows from related tables based on a condition
- LEFT JOIN – Return all rows from the left table and the matched rows from the right table
- RIGHT JOIN – Return all rows from the right table and the matched rows from the left table
- FULL JOIN – Return all rows when there's a match in either table
- CROSS JOIN – Return the Cartesian product of two tables
- SELF JOIN – Join a table to itself using aliases

Aggregating & Analyzing Data

- COUNT() – Count the number of rows
- SUM() – Calculate the sum of a column
- AVG() – Calculate the average of a column
- MIN() & MAX() – Find the minimum and maximum value in a column
- GROUP_CONCAT() – Concatenate values from a group

SQL Functions & Expressions

- COALESCE() – Return the first non-null value from a list
- NULLIF() – Return null if two expressions are equal
- CASE – Perform conditional logic in SQL queries
- CAST() – Convert a value to a specified data type
- CONCAT() – Concatenate two or more strings

Working with Tables

- CREATE TABLE – Create a new table
- ALTER TABLE – Modify an existing table
- DROP TABLE – Remove a table
- TRUNCATE TABLE – Remove all rows from a table without deleting the table structure
- RENAME TABLE – Rename a table

Constraints & Indexes

- PRIMARY KEY – Uniquely identify each row in a table
- FOREIGN KEY – Ensure referential integrity between two tables
- UNIQUE – Ensure unique values in a column
- CHECK – Ensure that all values in a column satisfy a condition
- DEFAULT – Set a default value for a column
- NOT NULL – Ensure a column cannot contain NULL values
- CREATE INDEX – Create an index on a table
- DROP INDEX – Remove an index

✨ **Views, Stored Procedures & Triggers**

- CREATE VIEW – Create a virtual table based on a SELECT statement
- DROP VIEW – Remove a view
- CREATE PROCEDURE – Create a stored procedure
- EXECUTE – Run a stored procedure
- DROP PROCEDURE – Remove a stored procedure
- CREATE TRIGGER – Create a trigger that executes a specified action when an event occurs
- DROP TRIGGER – Remove a trigger

User Management

- CREATE USER – Create a new user
- DROP USER – Remove a user
- ALTER USER – Change the password of a user
- GRANT – Give a user access to specific privileges
- REVOKE – Remove a user's access to specific privileges

Database Specific Commands

- **SQL Server:**
  - IDENTITY – Auto-increment a column's value
  - TOP – Limit the number of rows returned
- **Oracle**:
  - ROWNUM – Limit the number of rows returned
  - SEQUENCE – Create a sequence for generating unique numbers
  - NEXTVAL – Get the next value of a sequence

- **PostgreSQL:**
  - SERIAL – Auto-increment a column's value
  - LIMIT & OFFSET – Limit the number of rows returned and skip rows
- **MySQL:**
  - AUTO_INCREMENT – Auto-increment a column's value
  - LIMIT & OFFSET – Limit the number of rows returned and skip rows
  - SHOW TABLES – List all tables in the current database
  - DESCRIBE TABLE – Display a table's structure

NoSQL Cheat Sheet

NoSQL Database Types

- Key-Value – Stores data as key-value pairs (e.g., Redis, Riak)
- Document – Stores data as documents, typically in JSON format (e.g., MongoDB, Couchbase)
- Column-Family – Stores data in columns grouped together as column families (e.g., Cassandra, HBase)
- Graph – Stores data as nodes and edges in a graph (e.g., Neo4j, Amazon Neptune)

CRUD Operations

- CREATE – Add a new item to the database
- READ – Retrieve an item or items from the database
- UPDATE – Modify an existing item in the database
- DELETE – Remove an item from the database

MongoDB Specific

- db.collection.insertOne() – Insert a single document
- db.collection.insertMany() – Insert multiple documents
- db.collection.find() – Query documents
- db.collection.findOne() – Query a single document
- db.collection.updateOne() – Update a single document
- db.collection.updateMany() – Update multiple documents
- db.collection.deleteOne() – Delete a single document
- db.collection.deleteMany() – Delete multiple documents
- db.collection.createIndex() – Create an index

Cassandra Specific

- CREATE KEYSPACE – Create a new keyspace
- CREATE TABLE – Create a new table
- INSERT INTO – Insert data into a table

- `SELECT` – Retrieve data from a table
- `UPDATE` – Update data in a table
- `DELETE` – Remove data from a table
- `ALTER TABLE` – Modify an existing table
- `DROP TABLE` – Delete a table

Neo4j Specific

- `CREATE (n)` – Create a new node
- `CREATE (a)-[r:REL_TYPE]->(b)` – Create a new relationship between nodes
- `MATCH (n)` – Query nodes
- `MATCH (a)-[r:REL_TYPE]->(b)` – Query relationships
- `SET n.property = value` – Update a node's property
- `SET r.property = value` – Update a relationship's property
- `DETACH DELETE n` – Delete a node and its relationships
- `DELETE r` – Delete a relationship

Redis Specific

- `SET key value` – Set the value of a key
- `GET key` – Get the value of a key
- `DEL key` – Delete a key
- `EXISTS key` – Check if a key exists
- `INCR key` – Increment the integer value of a key
- `DECR key` – Decrement the integer value of a key
- `LPUSH key value` – Prepend a value to a list
- `RPUSH key value` – Append a value to a list
- `LPOP key` – Remove and return the first element of a list
- `RPOP key` – Remove and return the last element of a list
- `SADD key value` – Add a value to a set
- `SREM key value` – Remove a value from a set
- `SMEMBERS key` – Get all members of a set
- `HSET key field value` – Set the value of a field in a hash
- `HGET key field` – Get the value of a field in a hash
- `HDEL key field` – Delete a field from a hash
- `HGETALL key` – Get all fields and values of a hash
- `EXPIRE key seconds` – Set a key's time to live in seconds
- `TTL key` – Get the remaining time to live of a key

Couchbase Specific

- `CREATE BUCKET` – Create a new bucket
- `INSERT` – Insert a document into a bucket
- `SELECT` – Query documents using N1QL (SQL-like query language)
- `UPDATE` – Update a document

- DELETE – Remove a document from a bucket
- CREATE INDEX – Create an index for efficient querying
- DROP INDEX – Remove an index
- UPSERT – Insert or update a document

Amazon DynamoDB Specific

- CreateTable – Create a new table
- DeleteTable – Remove a table
- PutItem – Insert an item into a table
- UpdateItem – Modify an item in a table
- GetItem – Retrieve an item from a table
- DeleteItem – Remove an item from a table
- Query – Query items based on a condition
- Scan – Scan a table and retrieve items