SQL

* SQL = Structured Query Language.

SQL হচ্ছে ডাটাবেস অ্যাক্সেস এবং ম্যানিপুলেট করার জন্য একটি language.

এর মাধ্যমে database এ ডাটা insert, update, table create, এবং database থেকে ডাটা retrieve, delete এবং ডাটাবেস create করা যায়।

* Some of The Most Important SQL Commands
* SELECT - extracts data from a database
* UPDATE - updates data in a database
* DELETE - deletes data from a database
* INSERT INTO - inserts new data into a database
* CREATE DATABASE - creates a new database
* ALTER DATABASE - modifies a database
* CREATE TABLE - creates a new table
* ALTER TABLE - modifies a table
* DROP TABLE - deletes a table
* CREATE INDEX - creates an index (search key)
* DROP INDEX - deletes an index
* SQL Statement
* SELECT Statement: database থেকে ডাটা select করার জন্য SELECT Statement ব্যবহার করা হয়। ‍

Specific column select করার জন্য - SELECT CustomerName, City FROM Customers;

Table থেকে সকল ডাটা select করার জন্য - - SELECT \*FROM Customers;;

* DISTINCT Statement: database থেকে unique ডাটা return করে - SELECT DISTINCT name FROM `customers`।

WHERE clause: records কে filter করার জন্য ব্যবহার WHERE clause করা হয়। WHERE ক্লয়াজ ব্যবহার করা হয় ডেটাবেস থেকে নির্দিষ্ট তথ্য বা রেকর্ড রিট্রিভ করার জন্য, যা একটি বা একাধিক শর্তাদি প্রয়োগ করে সম্পাদিত হয়।

* ORDER BY Keyword: ডেটাবেস থেকে প্রাপ্ত রেকর্ডগুলি সর্ট করার জন্য। ORDER BY ডাটাকে ডিফল্ট ভাবে ascending order সর্ট করে।
* AND Operator: একাধিক condition উপর ভিত্তি করে ডেটার filter করার জন্য AND Operator ব্যবহার করা হয়। example - SELECT \* FROM `customers` WHERE email = 'm@gmail.com' and mobile = '66';
* OR Operator : একাধিক condition উপর ভিত্তি করে ডেটার filter করার জন্য OR Operator ব্যবহার করা হয়। example - SELECT \* FROM `customers` WHERE email = 'm@gmail.com' or mobile = '66';
* **NOT Operator:** **SQL NOT Operator** ব্যবহার করা হয় **ডাটাবেস থেকে নির্দিষ্ট ডেটা বাদ দিতে**। এটি একটি **logical operator** যা **WHERE** clause-এ ব্যবহার করা হয়। NOT operator-এর সাথে যেকোনো **Boolean expression** ব্যবহার করা যায়। example-

SELECT \* FROM `customers` WHERE id NOT IN(14,19);

NOT IN(), NOT BETWEEN, NOT LIKE, NOT IS NULL,

NOT Less Than- NOT NOT CustomerID < 50, NOT Greater Than- NOT CustomerID > 50

* NULL Value - SQL-এ NULL ভ্যালু হলো এমন একটা বিশেষ মান যা কোনো কলামে \*\* কোনো ডাটা নেই\*\* সেটা নির্দেশ করে। এটা কোনো ফাঁকা স্ট্রিং (empty string) বা শূণ্য (zero) এর মতো নয়। NULL মানের অর্থ হলো সেই কলামের জন্য কোনো তথ্য উপলব্ধ নেই বা প্রযোজ্য নয়।

এসকিউএলে, টেবিলের কলামগুলো ডাটা ধরে রাখে। কখনো কখনো কোনো কারণে কোনো নির্দিষ্ট রেকর্ডের জন্য সেই কলামে কোনো তথ্য নাও থাকতে পারে। এই অবস্থায়, সেই কলামের জন্য NULL মান সেট করা হয়।

* SQL UPDATE - SQL UPDATE statement হলো ডাটাবেসের টেবিলের মধ্যে  existing records পরিবর্তন করার জন্য ব্যবহৃত একটা কমান্ড. এটি ডাটা ম্যানিপুলেশন ল্যাংগুয়েজ (DML) এর অংশ, যা টেবিলের গঠন না বদলে কেবলমাত্র তথ্য পরিবর্তন করে।
* DELETE Statement - SQL DELETE হলো ডাটাবেসের মধ্যে  existing records থেকে ডাটা মুছে ফেলার জন্য ব্যবহৃত একটা SQL কমান্ড। এটি ডাটা ম্যানিপুলেশান ল্যাঙ্গুয়েজের (DML) অংশ, যা ডাটাবেসের ডাটা পরিবর্তন করতে ব্যবহৃত হয়।

DELETE কীভাবে কাজ করে?

DELETE কমান্ডটি নির্দিষ্ট টেবিল থেকে রেকর্ড মুছে ফেলতে ব্যবহার করা হয়। এটি একটি শর্তের উপর ভিত্তি করে নির্দিষ্ট রেকর্ড বা টেবিলের সবগুলি রেকর্ড মুছে ফেলতে পারে।

* **SQL LIMIT ক্লজ** - SQL ডেটাবেজ থেকে data সেটের আকার সীমাবদ্ধ করতে LIMIT ক্লজ ব্যবহার করা হয়। এটি প্রাথমিকভাবে নির্দিষ্ট করা সংখ্যক রো (Rows) return করে।
* SQL Aggregate Functions – একটি ডাটা সেট থেকে ক্যালকুলেশন করে একটি single value return করে।
* MIN() - returns the smallest value within the selected column
* MAX() - returns the largest value within the selected column
* COUNT() - returns the number of rows in a set
* SUM() - returns the total sum of a numerical column
* AVG() - returns the average value of a numerical column

COUNT() ফাংশন null value igone করে।

* SQL LIKE Operator - SQL LIKE অপারেটরটি কোনো টেবিলের কলামের মধ্যে কোনো নির্দিষ্ট প্যাটার্ন অনুসন্ধান করতে WHERE ক্লজের সাথে ব্যবহার করা হয়। এটি হল এমন একটি উপায় যা আপনাকে ঠিক সঠিক মিলের চেয়ে আরো নমনীয়ভাবে ডাটা খুঁজে বের করতে দেয়। LIKE অপারেটরটির সাথে সাথে বিশেষ কিছু অক্ষর ব্যবহার করা হয়, যা "wildcard" নামে পরিচিত। এই ওয়াইল্ডকার্ডগুলো আপনার নির্দিষ্ট করা প্যাটার্নের সাথে মেলে এমন একাধিক মানের সাথে মিল খুঁজতে সাহায্য করে।
* SQL Wildcard Characters - SQL ওয়াইল্ডকার্ড হল বিশেষ ক্যারেক্টার যা স্ট্রিং এর মধ্যে এক বা একাধিক ক্যারেক্টারের প্রতিস্থাপন হিসাবে ব্যবহৃত হয়। LIKE অপারেটরের সাথে এরা ব্যবহৃত হয় টেবিল থেকে ডাটা নির্বাচন করার সময় স্ট্রিং এর মধ্যে নির্দিষ্ট প্যাটার্ন অনুসন্ধান করতে।
* SQL IN Operator - SQL IN অপারেটর হলো WHERE cluse এ multiple ভালু নির্দিষ্ট করার জন্য ব্যবহৃত একটি way। এটি একাধিক OR কন্ডিশনের একটি সংক্ষিপ্ত রূপ।
* SQL BETWEEN Operator - SQL BETWEEN অপারেটর ডাটাবেসের টেবিল থেকে নির্দিষ্ট পরিসীমার মধ্যে থাকা ডাটা খুঁজে বের করতে ব্যবহৃত হয়। এটি WHERE ক্লজের সাথে কাজ করে এবং একটি সহজ উপায়ে ফলাফল ফিল্টার করতে দেয়।
* SQL aliases - হল সিকুয়াল টেবিলের নাম এবং কলাম নেইম কে করেন্ট অবস্থানে কিংবা স্থানীয় অংশের একটি নাম দেয়া। এই অ্যালিয়াসগুলি ব্যবহার করা হয় কুয়েরি পলকে কিংবা টেবিলের নাম লেখার সময় যাতে সংক্ষেপে ব্যবহার করা যায়। এটি কুয়েরি লেখার পার্থক্য তৈরি করে এবং সংক্ষেপে লেখা সুবিধা দেয়
* **SQL JOIN - SQL JOIN হলো এমন এক ধরণের ক্যuerী, যা দুটি বা তার বেশি টেবিলের ডাটা একত্রিত করে নতুন ডাটা সেট তৈরি করে। এই JOIN কাজ করে দুটি টেবিলে থাকা common** **INNER JOIN: এটি এমন JOIN, যা কেবল সেই রো (row) গুলোকেই ফলাফল দেয় যেগুলোর মধ্যে দুটি টেবিলেই মিল রয়েছে।**
* **LEFT JOIN: এটি এমন JOIN, যা বাম দিকের টেবিল থেকে সব রো (row) সহ, মিলি রয়েছে এমন ডান দিকের টেবিলের রো (row) গুলো ফলাফল দেয়। বাম দিকের টেবিলে মিল না থাকলে, ডান দিকের টেবিলে সেই কলামের জন্য NULL মান দেখাবে।**
* **RIGHT JOIN: এটি LEFT JOIN এর বিপরীত। এটি ডান দিকের টেবিল থেকে সব রো (row) সহ, মিলি রয়েছে এমন বাম দিকের টেবিলের রো (row) গুলো ফলাফল দেয়। বাম দিকের টেবিলে মিল না থাকলে, বাম দিকের টেবিলে সেই কলামের জন্য NULL মান দেখাবে।**
* **FULL JOIN: এটি এমন JOIN, যা দুটি টেবিলেই থাকা সমস্ত রো (row) ফলাফল দেয়, ভেবে মিল আছে কিনা।**
* **SQL সেলফ জয়েন - (Self Join) হলো একটি SQL কোয়েরি প্রযুক্তি যেখানে একটি টেবিলের সাথে নিজের নিজেকে জয়েন করা হয়। এটি মূলত একটি টেবিলের ভেতরের ডেটা সেটের মধ্যে সম্পর্ক নির্ধারণে ব্যবহৃত হয়। কলামের উপর ভিত্তি করে।**
* SQL UNION Operator - SQL UNION অপারেটর দুটি বা ততোধিক সিলেক্ট স্টেটমেন্টের output কে একটি সিঙ্গেল result সেটে মার্জ করে। UNION অপারেটর কেবল সমান সংখ্যক এবং সমান ধরনের কলামের মধ্যেই কাজ করতে পারে। প্রতিটি সিলেক্ট স্টেটমেন্টের ফলাফলের কলামের সংখ্যা, ধরণ এবং ক্রম সমমান হতে হবে।

SELECT name, price FROM `product\_two` UNION SELECT name, price FROM product\_one;

* **SQL GROUP BY Statement - GROUP BY ক্লজ SQL-এ একটি ডেটাবেস থেকে ডেটা গ্রুপ করার জন্য ব্যবহৃত হয়। এটি একই ধরণের ডেটা একত্রিত করে এবং প্রতিটি গ্রুপের জন্য output প্রদান করে।**

**SELECT price, sum(price) as "total price" FROM `product\_one` GROUP BY price;**

* **SQL HAVING Clause - HAVING clause ব্যবহার করা হয় GROUP BY clause দ্বারা গ্রুপ করা ডেটা সেটের উপর aggregate functions ফিল্টার করার জন্য। এটি WHERE clause এর মত কাজ করে, তবে WHERE clause ব্যবহার করে GROUP BY clause এর পূর্বেই ডেটা ফিল্টার করা হয়।**

**SELECT name,avg(price) FROM `product\_one` GROUP BY name HAVING sum(price) > 10;**

* **SQL EXISTS Operator - EXISTS অপারেটরটি একটি subquery ব্যবহার করে পরীক্ষা করে যে একটি টেবিলে একটি নির্দিষ্ট শর্ত পূরণ করে এমন কোন রেকর্ড আছে কিনা।**

**EXISTS** অপারেটরটি **subqueries** ব্যবহার করে ডেটা ফিল্টার করার জন্য একটি শক্তিশালী হাতিয়ার। এটি JOIN operations এর চেয়ে দ্রুত হতে পারে কারণ এটি subquery কে কেবলমাত্র প্রয়োজনীয় তথ্য খুঁজে পেতে ব্যবহার করে।

**EXISTS** অপারেটরটি কেবলমাত্র true বা false ফেরত দেয়। এটি কোন ডেটা ফেরত দেয় না।

**EXISTS** অপারেটরটি subqueries এর সাথে ব্যবহার করা হয় যা 1 রেকর্ড বা 0 রেকর্ড ফেরত দেয়।

**EXISTS** অপারেটরটি IN অপারেটরের বিকল্প হিসাবে ব্যবহার করা যেতে পারে।

**https://www.geeksforgeeks.org/sql-server-exists/**

* SQL ANY and ALL Operators-

**ANY এবং ALL** অপারেটরগুলো **SUBQUERY**-এর সাথে ব্যবহৃত হয়। SUBQUERY হলো একটি SQL প্রশ্ন যা অন্য SQL প্রশ্নের ভেতরে ব্যবহৃত হয়।

**ANY** অপারেটর **TRUE** retrun করে যদি SUBQUERY-এর ফলাফলে **কমপক্ষে একটি** মান SUBQUERY-এর শর্ত পূরণ করে।

**ALL** অপারেটর **TRUE** retrun করে যদি SUBQUERY-এর ফলাফলের **সকল** মান SUBQUERY-এর শর্ত পূরণ করে।

* **INSERT INTO SELECT Statement - INSERT INTO SELECT স্টেটমেন্ট ব্যবহার করে একটি টেবিলে ডেটা অন্য টেবিল থেকে কপি করা হয়।**
* **SQL CASE Expression - এটি ডেটাবেস থেকে রেকর্ড নির্বাচন করার সময়, শর্তের উপর নির্ভর করে বিভিন্ন মান ফেরত দিতে ব্যবহার করা হয়।**
* **একটি বা একাধিক শর্ত নির্ধারণ করা হয়।**
* **প্রতিটি শর্তের সাথে একটি মান যুক্ত করা হয়।**
* **একটি ডিফল্ট মান (ঐচ্ছিক) নির্ধারণ করা হয় যদি কোন শর্তই পূরণ না হয়।**
* **CASE Expression মূল্যায়ন করে এবং শর্তের সাথে মিলে যায় এমন প্রথম মান ফেরত দেয়।**
* **কোন শর্তই না মিললে ডিফল্ট মান ফেরত দেয়।**
* **IFNULL() এবং ISNULL() –**

**IFNULL(): expression1 যদি NULL হয়, তাহলে expression2 ফেরত দেয়। অন্যথায়, expression1 ফেরত দেয়।**

**IFNULL(expression1, expression2)**

**ISNULL(): expression1 যদি NULL হয়, তাহলে 1 ফেরত দেয়। অন্যথায়, expression1 ফেরত দেয়।**

**ISNULL(expression1)**

* **SQL Stored Procedures -** **SQL Stored Procedures হলো পূর্ব-সংজ্ঞায়িত কোডের ব্লক যা ডেটাবেজের সাথে ইন্টারঅ্যাক্ট করার জন্য ব্যবহৃত হয়। এগুলো SQL স্টেটমেন্টের সমন্বয়ে গঠিত হয় যা একসাথে সংকলিত হয় এবং একটি নাম দেওয়া হয়।** **Stored Procedures এর সুবিধা:**
* **কোড পুনরায় ব্যবহার: একই কাজ বারবার করার জন্য একই কোড বারবার লেখার প্রয়োজন হয় না।**