SQL Tutorial

1. **Getting information about database**

(1). Database name and version

SELECT version(), current\_date;

(2). Show time

SELECT now();

(3). Show all databases on the server

Show databases;

(4). Invoke a specific database (world)

use world;

(5). List tables in this database

Show tables;

(6). Invoke world\_x database

Use world\_x;

(7). List tables in this database

Show tables;

(8). Describe table

describe city;

(9). Show columns from table city

show columns from city

(10). Show create table

show create table city

1. **Simple outputs**

(1). Output simple Hello World

SELECT 'Hello, World';

(2). Simple addition

SELECT 1+3;

(3). Simple multiplication

SELECT 10\*30;

1. **SELECT rows(count, all, order by, offset)**

(1). Total number of countries

SELECT COUNT(\*) FROM country;

(2). All countries

SELECT \* FROM country;

(3). All countries in alphabetical order

SELECT \* FROM country ORDER BY Name;

(4). List the first 5 countries in alphabetical order

SELECT \* FROM country ORDER BY Name limit 5;

(5). List 10 countries from row 6 based on alphabetical order (OFFSET 5 will skip first 5 results of the query)

SELECT \* FROM country ORDER BY Name limit 10 offset 5;

1. **SELECT columns(as)**

(1). Country name and life expectancy rate

Use world

SELECT Name, LifeExpectancy FROM country;

(2). Country name ISO code, region and population

SELECT Name AS Country, Code AS ISO, Region, Population AS Pop FROM country;

(3). City name, country code and population

SELECT Name, CountryCode, Population from city;

(4). Country name, code, region and population/1000

SELECT Name AS Country, Code AS ISO, Region, Population / 1000 AS 'Population (1000s)' FROM country;

1. **SELECT with ORDER BY**

(1). Country name and life expectancy in descending order

SELECT Name, LifeExpectancy AS 'Life Expectancy' FROM country ORDER BY LifeExpectancy DESC;

(2). Country Name, ISO code, region and population ordered by ISO Code

SELECT Name AS Country, Code AS ISO, Region, Population FROM country ORDER BY Code;

(3). City name, country code and population ordered by city name

SELECT Name, CountryCode, Population from city ORDER BY Name;

(4). Country name, continent and region and order by continent, region and name

SELECT Name, Continent, Region FROM country ORDER BY Continent DESC, Region, Name;

1. **Working with WHERE**

(1). List of countries with population greater than 100 million

SELECT Name, Continent, Population FROM country WHERE Population > 100000000 ORDER BY Population DESC;

(2). List of countries that have population less than 100,000

SELECT Name, Continent, Population FROM country WHERE Population < 100000 ORDER BY Population DESC;

(3). List official languages of China

SELECT language FROM countrylanguage WHERE Countrycode = 'chn'

(4). select countries located in Asia

SELECT \* FROM country WHERE continent = 'Asia';

(5). select countries located in the Middle East

SELECT \* FROM country WHERE region = 'Middle East';

(6). select countries not located in the Middle East

SELECT \* FROM country WHERE region <> 'Middle East';

(7). select countries not located in Asia

SELECT \* FROM country WHERE continent != 'Asia';

(8). select count of countries located in Asia

SELECT count(\*) FROM country WHERE continent = 'Asia';

1. **Working with multiple tables(inner join, left join, right join)**

(1). List all countries and corresponding cities

SELECT co.Name as Country, c.Name as City

FROM country co, city c

WHERE co.Code = c.CountryCode

ORDER BY Country, city;

(2). INNER JOIN

SELECT co.Name as Country, c.Name as City

FROM country co

INNER JOIN city c on co.Code = c.CountryCode

ORDER BY Country, City;

1. **Working with LIKE and IN**

(1). Country names that start with United

SELECT Name, Continent, Population FROM country WHERE Name LIKE 'United%' ORDER BY Name;

(2). Country names that end with Island

SELECT Name, Continent, Population FROM country WHERE Name LIKE '%island' ORDER BY Name;

(3). All countries that are in Africa and North America continents

SELECT Name, Continent FROM country WHERE Continent IN ( 'Africa', 'North America' );

1. **Working with tables(create table, insert into, drop table)**

(1). Create new table with two columns Name and Age

CREATE TABLE People (Name Text, Age INT);

(2). Insert values to the new table

INSERT INTO People VALUES ('Michael', 35);

INSERT INTO People VALUES ('John', 15);

INSERT INTO People VALUES ('Ron', 55);

(3). Query the People table

SELECT \* FROM People;

(4). Create two new tables CUSTOMER and DRINK\_ORDER.

Notice how we create Primary and Foreign Keys and how we auto increment ID column.

CREATE TABLE customer (

id INTEGER auto\_increment,

first\_name VARCHAR(100),

last\_name VARCHAR(100),

address VARCHAR(100),

city VARCHAR(100),

state VARCHAR(2),

zip\_code VARCHAR(10),

PRIMARY KEY (id)

);

CREATE TABLE drink\_order (

id INTEGER auto\_increment,

customer\_id INTEGER,

drink\_description VARCHAR(100),

PRIMARY KEY (id),

CONSTRAINT fk\_drink\_order\_customer

FOREIGN KEY (customer\_id)

REFERENCES customer (id)

);

insert into customer values (null, 'Michael', 'Weston', '123 Brickel', 'Miami', 'FL', '33123');

insert into drink\_order values (123344, 1, 'Scotch');

select \* from customer;

select \* from drink\_order;

(5). Create table using existing table with SELECT (select count of all rows from an existing table COUNTRY; create new table COUNTRY2 using a SELECT statement, getting only country names which start with letter A; select from this new table COUNTRY2)

USE world

select count(\*) from country

CREATE TABLE country2 as (select \* from country where name like 'A%');

select \* from country2

(6). Dropping an existing table

Use world

Select \* from country2

DROP TABLE IF EXISTS country2

--you will get an error message here because we no longer have this table

select \* from country2

(7). Create table with indexes (Index will allow for faster retrieval of data)

Use world\_x

CREATE TABLE customer (

id INTEGER auto\_increment,

first\_name VARCHAR(100),

last\_name VARCHAR(100),

address VARCHAR(100),

city VARCHAR(100),

state VARCHAR(2),

zip\_code VARCHAR(10),

PRIMARY KEY (id),

INDEX last\_name\_idx (last\_name ASC)

);

CREATE TABLE drink\_order (

id INTEGER auto\_increment,

customer\_id INTEGER,

drink\_description VARCHAR(100),

PRIMARY KEY (id),

CONSTRAINT fk\_drink\_order\_customer

FOREIGN KEY (customer\_id)

REFERENCES customer (id)

);

insert into customer values (null, 'Michael', 'Weston', '123 Brickel', 'Miami', 'FL', '33123');

insert into drink\_order values (123344, 1, 'Scotch');

(8). Attempt to drop table with foreign keys dependencies

DROP TABLE If EXISTS customer

Should get an error code: Cannot delete or update a parent row: a foreign key constraint fails

Solution: first drop the table with constraint pointing to CUSTOMER table and then drop CUSTOMER TABLE

DROP TABLE IF EXISTS drink\_order

DROP TABLE IF EXISTS customer

1. **Updating**

(1). change Michael's age from 35 to 30

UPDATE People SET Age = 30 WHERE Name = 'Michael';

(2). Query the People table after the update --

SELECT \* FROM People;

(3). change Ron's name to Ronald and his age to 60--

UPDATE People SET Name = 'Ronald', Age = 60 WHERE Name = 'Ron';

(4). Query the People table after the update --

SELECT \* FROM People;

1. **Deleting**

(1). Delete John's record

DELETE FROM People WHERE Name = 'John';

(2). Query the People table after the delete -

SELECT \* FROM People;

1. **SELECT with AS**

(1).Select with aliases

SELECT code as country\_code, name as country\_name FROM country;

SELECT code as 'Country Code', name as 'Country Name' FROM country;

1. **SELECT with AND/OR clause**

(1). SELECT \* FROM country WHERE continent = 'Africa';

SELECT count(\*) FROM country WHERE continent = 'Africa';

SELECT \* FROM country WHERE continent = 'Africa' and region = 'Western Africa';

SELECT count(\*) FROM country WHERE continent = 'Africa' and region = 'Western Africa';

SELECT \* FROM country WHERE continent = 'Africa' or region = 'Central Africa'

SELECT \* FROM country WHERE continent = 'Africa' and region = 'Western Africa'

or continent = 'Africa' and region = 'Central Africa';

1. **SELECT with IN/NOT clauses**

(1).

SELECT \* FROM country WHERE continent = 'Europe';

SELECT \* FROM country WHERE continent IN ('Aisa', 'Africa');

SELECT count(\*) FROM country WHERE continent IN ('Aisa', 'Africa', 'South America');

SELECT count(\*) FROM country WHERE continent IN ('Aisa', 'Africa', 'South America')

AND region NOT IN ('Caribbean', 'Southern Europe')

1. **Functions**

(1). SUM

select SUM(salary) from employees;

(2). MIN/MAX

select MIN(salary) from employees;

SELECT MAX(salary)

FROM employees

WHERE job\_id = 'IT\_PROG';

(3). AVG

SELECT AVG(salary),count(\*)

FROM employees

WHERE department\_id = 90;

(4). Use all of them

SELECT ROUND(MAX(salary),0) 'Maximum',

ROUND(MIN(salary),0) 'Minimum',

ROUND(SUM(salary),0) 'Sum',

ROUND(AVG(salary),0) 'Average'

FROM employees;

(5). Show the difference between the highest and lowest salaries

SELECT MAX(salary) - MIN(salary) DIFFERENCE

FROM employees;

(6). LENGTH

SELECT \*

FROM employees

WHERE LENGTH(first\_name) >= 8;

(7). Write a query to list current date in the following format:

--Sample date: 2018-09-14

--Output date: September 17, 2019

SELECT DATE\_FORMAT(CURDATE(),'%M %e, %Y')

AS 'Current\_date';

--Sample date: 2018-09-14

--Output date: Tuesday September , 2019

SELECT DATE\_FORMAT(NOW(), '%W %M %Y');

(8). extract a year from a current date

SELECT EXTRACT(YEAR FROM NOW());

(9). LEFT/RIGHT

SELECT RIGHT('asdf', 1);

SELECT LEFT('asdf', 2);

SELECT FIRST\_NAME, LAST\_NAME,

concat(LEFT(FIRST\_NAME, 1), LEFT(LAST\_NAME, 1)) as Initials from employees;

1. **Group by**

(1). list the number of employees who have the same job

SELECT job\_id, COUNT(\*)

FROM employees

GROUP BY job\_id;

(2). list the department ID and the total salary paid in each department

SELECT department\_id, SUM(salary)

FROM employees

GROUP BY department\_id;

(3). list the average salary for each job ID EXLUDING programmer

SELECT job\_id, AVG(salary)

FROM employees

WHERE job\_id <> 'IT\_PROG'

GROUP BY job\_id;

(4). list the total salary, maximum, minimum, average salary of employees for department 90 only

SELECT job\_id, SUM(salary), AVG(salary), MAX(salary), MIN(salary)

FROM employees

WHERE department\_id = '90'

GROUP BY job\_id;

1. **JOIN**

(1). find the name (first\_name, last\_name), department id and name of all the employees

SELECT first\_name, last\_name, department\_id, department\_name

FROM employees

JOIN departments USING (department\_id);

(2). find the name (first\_name, last\_name), department id and name of all the employees who work in London

SELECT e.first\_name, e.last\_name, e.job\_id, e.department\_id, d.department\_name

FROM employees e

JOIN departments d

ON (e.department\_id = d.department\_id)

JOIN locations l ON

(d.location\_id = l.location\_id)

WHERE LOWER(l.city) = 'London';

(3). find the name (first\_name, last\_name) and hire date of the employees who were hired after ‘Jones’

SELECT e.first\_name, e.last\_name, e.hire\_date

FROM employees e

JOIN employees davies

ON (davies.last\_name = 'Jones')

WHERE davies.hire\_date < e.hire\_date;

(4). find employee id, the name (last\_name) along with their manager\_id and name (last\_name)

SELECT e.employee\_id 'Emp\_Id', e.last\_name 'Employee',

m.employee\_id 'Mgr\_Id', m.last\_name 'Manager'

FROM employees e

join employees m

ON (e.manager\_id = m.employee\_id);

1. **Subqueries**

(1). find a name (first\_name, last\_name) and salary of the employees who have a higher salary than the employee whose last\_name = ‘Bull’ (EMPLOYEES table from HR schema).

\*The subquery (inner query) will execute once before the main query (outer query) executes

\*The main query (outer query) will use the subquery result.

SELECT FIRST\_NAME, LAST\_NAME, SALARY

FROM employees

WHERE SALARY >

(SELECT salary FROM employees WHERE last\_name = 'Bull');

(2). find a name (first\_name, last\_name) of all employees who work in the IT department.

SELECT first\_name, last\_name

FROM employees

WHERE department\_id

IN (SELECT department\_id FROM departments WHERE department\_name='IT');

(3). find a name (first\_name, last\_name) of all employees who have a manager in USA based department. (EMPLOYEES table from HR schema).

SELECT first\_name, last\_name FROM employees

WHERE manager\_id in (select employee\_id

FROM employees WHERE department\_id

IN (SELECT department\_id FROM departments WHERE location\_id

IN (select location\_id from locations where country\_id='US')));

(4). find a name (first\_name, last\_name) and salary of the employees whose salary is greater than the average salary (EMPLOYEES table from HR schema).

SELECT first\_name, last\_name, salary FROM employees

WHERE salary > (SELECT AVG(salary) FROM employees);