SET 1

------

1. create a database called 'assignment' (Note please do the assignment tasks in this database)

Solution:

CREATE DATABASE assignment;

2. Use the file ConsolidatedTables.sql to create and populate the tables for the assignment .

3. Create a table called countries with the following columns

name, population, capital

- choose appropriate datatypes for the columns

Solution:

CREATE TABLE countries(

name varchar(255),

population integer,

capital varchar (255)

);

a) Insert the following data into the table

China 1382 Beijing

India 1326 Delhi

United States 324 Washington D.C.

Indonesia 260 Jakarta

Brazil 209 Brasilia

Pakistan 193 Islamabad

Nigeria 187 Abuja

Bangladesh 163 Dhaka

Russia 143 Moscow

Mexico 128 Mexico City

Japan 126 Tokyo

Philippines 102 Manila

Ethiopia 101 Addis Ababa

Vietnam 94 Hanoi

Egypt 93 Cairo

Germany 81 Berlin

Iran 80 Tehran

Turkey 79 Ankara

Congo 79 Kinshasa

France 64 Paris

United Kingdom 65 London

Italy 60 Rome

South Africa 55 Pretoria

Myanmar 54 Naypyidaw

Solution:

insert into countries values('China',1382,'Beijing'),

('India',1326,'Delhi'),

('United States',324,'Washington D.C.'),

('Indonesia',260,'Jakarta'),

('Brazil',209,'Brasilia'),

('Pakistan',193,'Islamabad'),

('Nigeria',187,'Abuja'),

('Bangladesh',163,'Dhaka'),

('Russia',143,'Moscow'),

('Mexico',128,'Mexico City'),

('Japan',126,'Tokyo'),

('Philippines',102,'Manila'),

('Ethiopia',101,'Addis Ababa'),

('Vietnam',94,'Hanoi'),

('Egypt',93,'Cairo'),

('Germany',81,'Berlin'),

('Iran',80,'Tehran'),

('Turkey',79,'Ankara'),

('Congo',79,'Kinshasa'),

('France',64,'Paris'),

('United Kingdom',65,'London'),

('Italy',60,'Rome'),

('South Africa',55,'Pretoria'),

('Myanmar',54,'Naypyidaw');

b) Add a couple of countries of your choice

Solution: Can be any values of their choice

Example

insert into countries values('Thailand',68,'Bangkok');

c) Change ‘Delhi' to ‘New Delhi'

Solution:

update countries set capital = ‘New Delhi' where capital = 'Delhi';

d) Delete Ethiopia and Vietnam from the table

delete from countries where name IN (‘Ethiopia’,’Vietnam’);

4. Rename the table countries to big\_countries .

Solution 1 :

alter table countries rename to big\_countries;

or

rename table countries to big\_countries;

5. Create the following tables. Use auto increment wherever applicable

a. Product

product\_id - primary key

product\_name - cannot be null and only unique values are allowed

description

supplier\_id - foreign key of supplier table

b. Suppliers

supplier\_id - primary key

supplier\_name

location

c. Stock

id - primary key

product\_id - foreign key of product table

balance\_stock

Solution:

create table suppliers(

suppllier\_id integer primary key auto\_increment,

supplier\_name varchar(30),

location varchar(50)

);

create table product (

product\_id integer primary key auto\_increment,

product\_name varchar(50) NOT NULL UNIQUE,

description varchar(150),

supplier\_id integer,

foreign key(supplier\_id) references suppliers(supplier\_id)

);

create table stock(

id integer primary key auto\_increment,

product\_id integer,

balance\_stock integer,

foreign key(product\_id) references products(product\_id)

);

6. Enter some records into the three tables.

Solution: They can insert any value of their choice

Example :

insert into suppliers values (1,'New India Suppliers','Mumbai');

insert into suppliers values (2, 'Nav Bharath Suppliers','Delhi');

insert into product values(100,'Wet Grinder', 'Heavy duty wet grinders', 1),

(200,'Sewing Machine', ' Export quality sewing machines with 23 settings',2);

insert into stock values (1,100,23),(2,200,5);

7. Modify the supplier table to make supplier name unique and not null.

Solution:

alter table suppliers modify column supplier\_name varchar(30) UNIQUE NOT NULL;

8. Modify the emp table as follows

a. Add a column called deptno

Solution:

alter table emp add column deptno integer;

b. Set the value of deptno in the following order

deptno = 20 where emp\_id is divisible by 2

deptno = 30 where emp\_id is divisible by 3

deptno = 40 where emp\_id is divisible by 4

deptno = 50 where emp\_id is divisible by 5

deptno = 10 for the remaining records.

Solution:

update emp set deptno = 10;

update emp set deptno = 20 where emp\_id%2 = 0;

update emp set deptno = 30 where emp\_id%3 = 0;

update emp set deptno = 40 where emp\_id%4 = 0;

update emp set deptno = 50 where emp\_id%5 = 0;

9. Create a unique index on the emp\_id column.

Solution:

create unique index hidx on emp(emp\_id);

10 Create a view called ***emp\_sal*** on the **emp** table by selecting the following fields in the order of highest salary to the lowest salary.

emp\_no, first\_name, last\_name, salary

Solution:

create view emp\_sal as select emp\_no, first\_name, last\_name, salary from emp order by salary desc;

SET 2

1. select all employees in department 10 whose salary is greater than 3000. [table: employee]

Solution:

select \* from employee where salary > 3000;

2. The grading of students based on the marks they have obtained is done as follows:

40 to 50 -> Second Class

50 to 60 -> First Class

60 to 80 -> First Class

80 to 100 -> Distinctions

a. How many students have graduated with first class?

Solution:

select count(\*) from students where marks between 50 and 60;

+----------+

| count(\*) |

+----------+

| 4 |

+----------+

1 row in set (0.000 sec)

b. How many students have obtained distinction? [table: students]

Solution

select count(\*) from students where marks between 80 and 100;

+----------+

| count(\*) |

+----------+

| 11 |

+----------+

1 row in set (0.000 sec)

3. Get a list of city names from station with even ID numbers only. Exclude duplicates from your answer.[table: station]

Solution:

select distinct city from station where id%2=0;

4. Find the difference between the total number of city entries in the table and the number of distinct city entries in the table. In other words, if N is the number of city entries in station, and N1 is the number of distinct city names in station, write a query to find the value of N-N1 from station.

[table: station]

Solution:

select count(city) - count(distinct city) as Difference from station;

+------------+

| Difference |

+------------+

| 30 |

+------------+

1 row in set (0.001 sec)

5. a. Query the list of CITY names starting with vowels (i.e., a, e, i, o, or u) from STATION. Your result cannot contain duplicates. [Hint: Use RIGHT() / LEFT() methods ]

Solution:

select distinct city from station where RIGHT(city,1) in ('a','e','i','o','u');

b. Query the list of CITY names from STATION which have vowels (i.e., a, e, i, o, and u) as both their first and last characters. Your result cannot contain duplicates.

Solution:

select distinct city from station where RIGHT(city,1) in ('a','e','i','o','u') and LEFT(city,1) in ('a','e','i','o','u');

c. Query the list of CITY names from STATION that do not start with vowels. Your result cannot contain duplicates.

Solution:

select distinct city from station where RIGHT(city,1) not in ('a','e','i','o','u');

d. Query the list of CITY names from STATION that either do not start with vowels or do not end with vowels. Your result cannot contain duplicates. [table: station]

Solution:

select distinct city from station where RIGHT(city,1) in ('a','e','i','o','u') or LEFT(city,1) in ('a','e','i','o','u');

6. Write a query that prints a list of employee names having a salary greater than $2000 per month who have been employed for less than 20 months. Sort your result by ascending emp\_id. [table: emp]

Solution:

select \* from emp where date\_sub(date(now()), INTERVAL 36 MONTH) < hire\_date and salary >5000;

7. How much money does the company spend every month on salaries for each department? [table: employee]

Solution:

select deptno, sum(salary) from employee group by deptno;

+--------+-------------+

| deptno | sum(salary) |

+--------+-------------+

| 10 | 20700.00 |

| 20 | 12300.00 |

| 30 | 1675.00 |

+--------+-------------+

3 rows in set (0.002 sec)

8. How many cities in the CITY table have a Population larger than 100000. [table: city]

Solution:

select count(\*) from city where population > 100000;

+----------+

| count(\*) |

+----------+

| 11 |

+----------+

1 row in set (0.001 sec)

9. What is the total population of California? [table: city]

Solution:

select sum(population) from city where district = 'california';

+-----------------+

| sum(population) |

+-----------------+

| 339002 |

+-----------------+

1 row in set (0.000 sec)

10. What is the average population of the districts in each country? [table: city]

Solution:

select countrycode, avg(population) as AvgPopulation from city group by countrycode;

+-------------+---------------+

| countrycode | AvgPopulation |

+-------------+---------------+

| JPN | 175839.2000 |

| NLD | 593321.0000 |

| USA | 120225.8750 |

+-------------+---------------+

3 rows in set (0.000 sec)

11. Find the ordernumber, status, customernumber, customername and comments for all orders that are ‘Disputed= [table: orders, customers]

Solution

select o.ordernumber, o.status, o.comments, c.customernumber, c.customername from orders o inner join customers c on c.customernumber = o.customernumber

where status='Disputed';

SET 3

-----

1. Write a stored procedure that accepts the month and year as inputs and prints the ordernumber, orderdate and status of the orders placed in that month.

***Example***: call order\_status(2005, 11);

Solution:

CREATE PROCEDURE `order\_status`(IN ord\_year integer, IN ord\_month integer)

BEGIN

select ordernumber, orderdate, status

from orders

where year(orderdate) = ordyear

and month(orderdate) = lclmonth ;

END;

2. Write a stored procedure to insert a record into the cancellations table for all cancelled orders.

STEPS:

a. Create a table called cancellations with the following fields

id (primary key),

customernumber (foreign key - Table customers),

ordernumber (foreign key - Table Orders),

comments

All values except id should be taken from the order table.

Solution:

create table cancellations (

id integer primary key auto\_increment,

customernumber integer ,

ordernumber integer,

foreign key(customernumber) references customers(customernumber),

foreign key(ordernumber) references orders(ordernumber));

b. Read through the orders table record by record. If an order is cancelled, then put an entry in the cancellations table.

Solution:

CREATE DEFINER=`root`@`localhost` PROCEDURE `proc\_cancel`()

BEGIN

declare cnum, ordnum, finished integer default 0;

-- Declare Cursor

declare ord\_cur cursor for

select customernumber, ordernumber

from orders

where status='cancelled';

-- Declare exception handler

declare exit handler for NOT FOUND set finished = 1;

-- Open Cursor

open ord\_cur;

-- Fetch order records

ordloop:REPEAT

fetch ord\_cur into cnum, ordnum;

insert into cancellations (customernumber, ordernumber) values(cnum, ordnum);

until finished = 1

end repeat ordloop;

END;

3. a. Write function that takes the customernumber as input and returns the purchase\_status based on the following criteria . [table:Payments]

if the total purchase amount for the customer is < 25000 , then status = Silver,

if amount is between 25000 and 50000, then status = Gold

if amount > 50000 then status = Platinum

Solution:

CREATE FUNCTION `purchase\_status`(cnum integer)

RETURNS varchar(10)

BEGIN

declare pstatus varchar(10) default '';

declare amt integer default 0;

select sum(amount) into amt from payments where customerNumber = cnum;

if amt < 25000 then

set pstatus = 'Silver';

elseif amt between 25000 and 50000 then

set pstatus = 'Gold';

else

set pstatus = 'Platinum';

end if;

RETURN pstatus;

END

b. Write a query that displays customerNumber, customername and purchase\_status from customers table.

Solution:

select customerNumber, customerName, purchase\_status(customerNumber) from customers;

4. Replicate the functionality of 'on delete cascade' and 'on update cascade' using triggers on movies and rentals tables.

Note: Both tables - movies and rentals - don't have primary or foreign keys. Use only triggers to implement the above.

Solution

-- On update cascade

CREATE TRIGGER upd\_cascade

AFTER UPDATE ON `movies`

FOR EACH ROW

BEGIN

update rentals set movieid = new.id

where movieid = old.id;

END;

-- On delete cascade

CREATE TRIGGER del\_cascade

AFTER DELETE ON `movies`

FOR EACH ROW

BEGIN

delete from rentals

where movieid = old.id;

END;

5. Select the first name of the employee who gets the third highest salary. [table: employee]

Solution

SELECT fname, lname, salary,

NTH\_VALUE(fname, 3) OVER ( ORDER BY salary DESC) third\_highest\_salary

FROM employee;

6. Assign a rank to each employee based on their salary. The person having the highest salary has rank 1. [table: employee]

Solution

SELECT fname, lname, salary,

DENSE\_RANK() OVER ( ORDER BY salary DESC ) emp\_rank

FROM employee;