**Day 9 Assignments 1&2:**

**Assignment 1:**

**Design a database schema for a library system, including tables, fields, and constraints like NOT NULL,UNIQUE, and CHECK. Include primary and foreign keys to establish relationships between table.**

> CREATE TABLE Writers

(

writer\_id INT PRIMARY KEY AUTO\_INCREMENT,

writer\_name VARCHAR(50) NOT NULL

);

> CREATE TABLE Publications

(

publication\_id INT PRIMARY KEY AUTO\_INCREMENT,

title VARCHAR(50) NOT NULL,

writer\_id INT,

isbn VARCHAR(20) UNIQUE,

pub\_year INT,

summary TEXT,

FOREIGN KEY (writer\_id) REFERENCES Writers(writer\_id)

);

**Assignment 2: practice q/ns**

CREATE TABLE Manufacturers

(

Code INTEGER,

Name VARCHAR(255) NOT NULL,

PRIMARY KEY (Code)

);

CREATE TABLE Products (

Code INTEGER,

Name VARCHAR(255) NOT NULL ,

Price DECIMAL NOT NULL ,

Manufacturer INTEGER NOT NULL,

PRIMARY KEY (Code),

FOREIGN KEY (Manufacturer) REFERENCES Manufacturers(Code)

) ENGINE=INNODB;

INSERT INTO Manufacturers(Code,Name) VALUES(1,'Sony');

INSERT INTO Manufacturers(Code,Name) VALUES(2,'Creative Labs');

INSERT INTO Manufacturers(Code,Name) VALUES(3,'Hewlett-Packard');

INSERT INTO Manufacturers(Code,Name) VALUES(4,'Iomega');

INSERT INTO Manufacturers(Code,Name) VALUES(5,'Fujitsu');

INSERT INTO Manufacturers(Code,Name) VALUES(6,'Winchester');

INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(1,'Hard drive',240,5);

INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(2,'Memory',120,6);

INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(3,'ZIP drive',150,4);

INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(4,'Floppy disk',5,6);

INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(5,'Monitor',240,1);

INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(6,'DVD drive',180,2);

INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(7,'CD drive',90,2);

INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(8,'Printer',270,3);

INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(9,'Toner cartridge',66,3);

INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(10,'DVD burner',180,2);

**1. Select the names of all the products in the store.**

>select Name from Products;

**2. Select the names and the prices of all the products in the store.**

>select name, price from products;

**3. Select the name of the products with a price less than or equal to $200.**

>select name from products where price <= 200;

**4. Select all the products with a price between $60 and $120.**

>select \* from products where price between 60 and 120;

>select \* from products where price >= 60 and price <= 120;

**5. Select the name and price in cents (i.e., the price must be multiplied by 100).**

>select name, price\*100 from products;

>select name, concat(price\*100, ' cents') from products;

**6. Compute the average price of all the products.**

>select avg(price) from products;

>select sum(price)/count(price) from products;

**7. Compute the average price of all products with manufacturer code equal to 2.**

>select avg(price) from products where Manufacturer = 2;

**8. Compute the number of products with a price larger than or equal to $180.**

>select count(\*) from products where price>=180;

**9. Select the name and price of all products with a price larger than or equal to $180, and sort first by price (in descending order), and then by name (in ascending order).**

>select name, price from products where price>=180 order by price desc, name asc;

**10. Select all the data from the products, including all the data for each product's manufacturer.**

>select a.\*, b.name from products a join Manufacturers b on(a.manufacturer = b.code);

>select a.\*, b.name from products a, Manufacturers b where a.manufacturer = b.code;

**11. Select the product name, price, and manufacturer name of all the products.**

>select a.name, a.price, b.name from products a join Manufacturers b on(a.manufacturer = b.code);

>SELECT Products.Name, Price, Manufacturers.Name

FROM Products INNER JOIN Manufacturers

ON Products.Manufacturer = Manufacturers.Code;

**12. Select the average price of each manufacturer's products, showing only the manufacturer's code.**

>SELECT AVG(Price), Manufacturer FROM Products

GROUP BY Manufacturer;

**13. Select the average price of each manufacturer's products, showing the manufacturer's name.**

>select avg(a.price), b.name

from Products a join Manufacturers b

on a.manufacturer = b.code

group by b.name;

**14. Select the names of manufacturer whose products have an average price larger than or equal to $150.**

>select avg(a.price), b.name

from Manufacturers b join Products a

on b.code = a.Manufacturer

group by b.name

having avg(a.price)>=150;

>SELECT AVG(Price), Manufacturers.Name

FROM Products, Manufacturers

WHERE Products.Manufacturer = Manufacturers.Code

GROUP BY Manufacturers.Name

HAVING AVG(Price) >= 150;

**15. Select the name and price of the cheapest product.**

>select name, price from Products

where price = (

select min(price)

from products);

>SELECT name,price

FROM Products

ORDER BY price ASC

LIMIT 1;

🡪SQL SERVER SOLUTION (T-SQL)

>SELECT TOP 1 name, price

FROM Products

ORDER BY price ASC

**16. Select the name of each manufacturer along with the name and price of its most expensive product.**

>select max\_price\_mapping.name as manu\_name, max\_price\_mapping.price, products\_with\_manu\_name.name as product\_name

from

(SELECT Manufacturers.Name, MAX(Price) price

FROM Products, Manufacturers

WHERE Manufacturer = Manufacturers.Code

GROUP BY Manufacturers.Name)

as max\_price\_mapping

left join

(select products.\*, manufacturers.name manu\_name

from products join manufacturers

on (products.manufacturer = manufacturers.code))

as products\_with\_manu\_name

on

(max\_price\_mapping.name = products\_with\_manu\_name.manu\_name

and

max\_price\_mapping.price = products\_with\_manu\_name.price);

**17. Add a new product: Loudspeakers, $70, manufacturer 2.**

>insert into Products values (11, 'Loudspeakers', 70, 2);

**18. Update the name of product 8 to "Laser Printer".**

>update products

set name = 'Laser Printer'

where code=8;

**19. Apply a 10% discount to all products.**

>update products

set price=price\*0.9;

**20. Apply a 10% discount to all products with a price larger than or equal to $120.**

>update products

set price = price \* 0.9

where price >= 120;