**Week – 7**

**Hands on Lab Questions**

1. **Table: Students**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Student\_ID** | **Firstname** | **Lastname** | **Age** | **Gender** | **Grade** |  |
| 1 | Rakesh | Shrestha | 19 | Male | A |
| 2 | Binita | Bhandari | 21 | Female | B |
| 3 | Bobby | Sherpa | 20 | Male | C |
| 4 | Sagar | Parajuli | 22 | Male | A |
| 5 | Prajina | KC | 21 | Female | B |

Queries:

CREATE TABLE `workshop 7`.`students` (

    `Student\_ID` INT NOT NULL,

    `Firstname` VARCHAR(20) NOT NULL,

    `Lastname` VARCHAR(15) NOT NULL,

    `Age` INT NOT NULL,

    `Gender` VARCHAR(10) NOT NULL,

    `Grade` VARCHAR(1) NOT NULL

) ENGINE = InnoDB;

ALTER TABLE students

MODIFY Student\_ID INT AUTO\_INCREMENT PRIMARY KEY;

INSERT INTO `students` (

        `Student\_ID`,

        `Firstname`,

        `Lastname`,

        `Age`,

        `Gender`,

        `Grade`

    )

VALUES (NULL, 'Rakesh', 'Shrestha', '19', 'Male', 'A'),

    (NULL, 'Binita', 'Bhandari', '21', 'Female', 'B'),

    (NULL, 'Bobby', 'Sherpa', '20', 'Male', 'C'),

    (NULL, 'Sagar', 'Parajuli', '22', 'Male', 'A'),

    (NULL, 'Prajina', 'KC', '21', 'Female', 'B')

* 1. Retrieve all data from the Students Table

SELECT \* FROM `students`

Table

Description automatically generated with low confidence

* 1. Retrieve first name and last name of all students

SELECT Firstname, Lastname FROM `students`

Graphical user interface, application

Description automatically generated

* 1. Retrieve only the students who have a grade A

SELECT Firstname, Lastname FROM students WHERE Grade = 'A'

Graphical user interface, application

Description automatically generated

* 1. Retrieve only the students who are older than 20

SELECT \* FROM `students` WHERE Age >= 20;

Graphical user interface, application

Description automatically generated

* 1. Retrieve the count of male and female students

SELECT COUNT(CASE WHEN Gender='Male' THEN 1 END),

COUNT(CASE WHEN Gender='Female' THEN 1 END) FROM students

Graphical user interface, text, application

Description automatically generated

* 1. Update the grade of a specific student (e.g. StudentID=3)

UPDATE `students` SET Grade = 'A' WHERE Student\_ID = 3;

* 1. Increment the age of all male students by 1

UPDATE `students` SET Age = Age+1;

Graphical user interface, application

Description automatically generated

* 1. Update the grade of all female students who have a grade of 'B' to 'A'

UPDATE `students` SET Grade = 'A' WHERE Grade = 'B';

Graphical user interface, application

Description automatically generated

1. **Table: Customers**

**Customer\_ID Firstname Lastname Age Gender Email**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | Kabir | Sharma | 25 | Male | kabir@gmail.com |
| 2 | Haris | Shrestha | 30 | Male | haris@gmail.com |
| 3 | Deepti | Singh | 26 | Female | deepti@gmail.com |

**Table: Orders**

**OrderID CustomerID OrderDate Total**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 1 | 2023-04-06 | 1000 |
| 2 | 1 | 2023-04-01 | 3000 |
| 3 | 2 | 2023-03-22 | 3500 |
| 4 | 3 | 2023-04-04 | 2300 |

**Table: OrderItems**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | 3 | Product D | 5 | 950 |
| 5 | 3 | Product E | 1 | 1125 |
| 6 | 4 | Product F | 2 | 1780 |
| 7 | 4 | Product G | 1 | 1560 |
| 8 | 3 | Product H | 3 | 1220 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OrderItemID** | **OrderID** | **ProductItems** | **Quantity** | **Price** |
| 1 | 1 | Product A | 2 | 1200 |
| 2 | 1 | Product B | 3 | 1345 |
| 3 | 2 | Product C | 4 | 1360 |

Queries:

CREATE TABLE customers(

    Customer\_ID INT PRIMARY KEY AUTO\_INCREMENT,

    Firstname VARCHAR(20) NOT NULL,

    Lastname VARCHAR(20) NOT NULL,

    Age INT NOT NULL,

    Gender VARCHAR(10) NOT NULL,

    email INT

);

CREATE TABLE orders (

    ORDER\_ID INT PRIMARY KEY AUTO\_INCREMENT,

    Customer\_ID INT NOT NULL,

    `Order\_date` DATE NOT NULL,

    `Total` INT NOT NULL

);

ALTER TABLE orders

ADD CONSTRAINT fk\_orders\_customers FOREIGN KEY (Customer\_ID) REFERENCES customers(Customer\_ID);

INSERT INTO `customers` (

        `Customer\_ID`,

        `Firstname`,

        `Lastname`,

        `Age`,

        `Gender`,

        `email`

    )

VALUES (

        NULL,

        'Kabir',

        'Sharma',

        '25',

        'Male',

        'kabir@gmail.com'

    ),

    (

        NULL,

        'Haris',

        'Shrestha',

        '30',

        'Male',

        'haris@gmail.com'

    ),

    (

        NULL,

        'Deepti',

        'Singh',

        '26',

        'Female',

        'deepti@gmail.com'

    )

INSERT INTO `orders` (`ORDER\_ID`, `Customer\_ID`, `Order\_date`, `Total`)

VALUES (NULL, '1', '2023-04-06', '1000'),

    (NULL, '1', '2023-04-01', '3000'),

    (NULL, '2', '2023-03-22', '3500'),

    (NULL, '3', '2023-04-04', '2300') CREATE TABLE `workshop 7`.`orderitems` (

        `OrderItemID` INT NOT NULL,

        `OrderID` INT NOT NULL,

        `ProductItems` VARCHAR(15) NOT NULL,

        `Quantity` INT NOT NULL,

        `Price` INT NOT NULL

    ) ENGINE = InnoDB;

ALTER TABLE orderitems

ADD CONSTRAINT fk\_orderitems\_orders FOREIGN KEY (OrderID) REFERENCES orders(ORDER\_ID);

ALTER TABLE orderitems

MODIFY OrderItemID INT AUTO\_INCREMENT PRIMARY KEY;

INSERT INTO `orderitems` (`OrderID`, `ProductItems`, `Quantity`, `Price`)

VALUES ('1', 'Product A', '2', '1200'),

    ('1', 'Product B', '3', '1345'),

    ('2', 'Product C', '4', '1360'),

    ('3', 'Product D', '5', '950'),

    ('3', 'Product E', '1', '1125'),

    ('4', 'Product F', '2', '1780'),

    ('4', 'Product G', '1', '1560'),

    ('3', 'Product H', '3', '1220');

Graphical user interface, application

Description automatically generated

1. Retrieve all orders along with the customer details

SELECT \*

FROM orders o

    JOIN customers c USING(Customer\_ID);

Graphical user interface, application

Description automatically generated

1. Retrieve all orders along with the customer details and order items

SELECT \*

FROM orders o

JOIN customers c USING(Customer\_ID)

JOIN orderitems oi USING(OrderID);

Graphical user interface, table

Description automatically generated

1. Retrieve the customer details for a specific order (e.g. OrderID=2)

SELECT \*

FROM orders o

JOIN customers c ON o.Customer\_ID = c.Customer\_ID

WHERE o.ORDER\_ID = 2;

Graphical user interface, application

Description automatically generated

1. Retrieve all orders for a specific customer (e.g. CustomerID=1)

SELECT \*

FROM orders o

JOIN customers c ON o.Customer\_ID = c.Customer\_ID

WHERE o.Customer\_ID = 1;

Graphical user interface, text, application

Description automatically generated

1. Retrieve the total revenue generated by each customer

SELECT c.Customer\_ID,

    c.Firstname,

    c.Lastname,

    SUM(oi.Price) AS TotalRevenue

FROM customers c

    JOIN orders o ON o.Customer\_ID = c.Customer\_ID

    JOIN orderitems oi ON oi.OrderID = o.ORDER\_ID

GROUP BY c.Customer\_ID;

Graphical user interface, application

Description automatically generated