

```
//UQ1
```

```
INSERT INTO Customer (customer_name, street_address, zip_code, email,  
phone)  
VALUES ('John Smith', '123 Drury lane', 79111, 'JSmith@cxr.com', '310-805-  
9801');
```

```
//UQ2
```

```
UPDATE CarDetails  
SET mileage = mileage +10000  
WHERE vin = '1N6AD0CU7EN585013'
```

```
//UQ3
```

```
INSERT INTO TicketDetails (ticket_date, customer_id, vin, issue_desc)  
VALUES (CURRENT_DATE, 1, '3D7TP2CT0AG772172', 'Change Transmission  
fluid');
```

```
//UQ4
```

```
Update PartInventory  
SET available_quantity = -10  
WHERE part_id = '1000'
```

-- IQ1- How many customers with balance over \$50?

```
CREATE VIEW custbalance AS  
    SELECT Customer_ID, Remaining_Balance,  
           IF(Remaining_Balance < 50, null, Remaining_Balance) as itemCost  
    FROM billingdetails;  
SELECT count(itemCost) FROM custbalance;
```

-- IQ2- List all customer names, bills numbers, and each bill balance if the bill balance >0

-- drop view custnameswbalance;

```
CREATE VIEW custnameswbalance AS  
    SELECT customer_name, bill_number,  
           IF(bill_total > 0, null, bill_total) as custbal  
    FROM Customer cross join BillingDetails;
```

```
SELECT *  
from custnameswbalance;
```

-- IQ3- List the history of the car with vin = '4USBU53567L660273'. The history must include all tickets information, associated services, and replaced parts' numbers, descriptions, and used quantities

```

CREATE VIEW Vin_Hist AS
SELECT ticketdetails.ticket_id, ticketdetails.ticket_date, ticketdetails.issue_desc, ticketdetails.vin,
partsneeded.part_id, partsneeded.quantity_used, partinventory.part_desc
FROM ticketdetails
JOIN servicedetails
ON ticketdetails.ticket_id=servicedetails.ticket_id
JOIN partsneeded
ON servicedetails.service_id=partsneeded.service_id
JOIN partinventory
ON partsneeded.part_id=partinventory.part_id
WHERE vin = '4USB053567L660273';
SELECT * from Vin_Hist;

```

-- IQ4- Given part number ('390'), List information of all tickets associated with a service that used that part and the used quantity

```

CREATE VIEW Part_Hist AS
SELECT partsneeded.quantity_used, partsneeded.part_id, partsneeded.service_id,
billingdetails.completed_services, billingdetails.ticket_id
FROM partsneeded
JOIN billingdetails
ON partsneeded.service_id=billingdetails.service_id
WHERE partsneeded.part_id = 390;
SELECT * from Part_Hist;

```

-- IQ5 How many tickets have been made in the interval between the two dates ('2015-01-06', '2015-02-07')?

```

CREATE VIEW Ticket_Interval AS
SELECT COUNT(ticket_id)
FROM ticketdetails
WHERE ticket_date BETWEEN '2015-01-06' and '2015-02-07';
SELECT * from Ticket_Interval;

```

-- IQ6- what is the total dollar values (labor and parts) associated with tickets in the previous query (IQ5)?

```

CREATE VIEW Dollar_Total_View AS
SELECT SUM(labor_cost + total_parts_cost)
FROM billingdetails
INNER JOIN ticketdetails
ON billingdetails.ticket_id=ticketdetails.ticket_id
WHERE ticket_date BETWEEN '2015-01-06' and '2015-02-07';

```

```
SELECT * from Dollar_Total_View;
```

-- IQ7- Given a part ID ('1211'), find the information of all suppliers (vendors) for that part?

```
CREATE VIEW Part_Supplier_View AS
```

```
SELECT availableparts.part_id, availableparts.vendor_id, vendor.vendor_name, vendor.vendor_phone  
FROM availableparts
```

```
JOIN vendor
```

```
ON availableparts.vendor_id=vendor.vendor_id
```

```
WHERE availableparts.part_id = 1211;
```

```
SELECT * from Part_Supplier_View;
```

-- IQ8- What is the part with highest demand?

```
CREATE VIEW Highest_Demand_View AS
```

```
SELECT MAX(quantity_used) AS HighestDemand, part_id  
FROM partsneeded;
```

```
SELECT * from Highest_Demand_View;
```

-- IQ9 -What is the smallest, largest and average bill total?

```
CREATE VIEW Small_Large_View AS
```

```
SELECT MIN(bill_total) AS SmallestPrice  
FROM billingdetails;
```

```
SELECT MAX(bill_total) AS LargestPrice  
FROM billingdetails;
```

```
SELECT AVG(bill_total)  
FROM billingdetails;
```

```
SELECT * from Small_Large_View;
```

-- IQ10-What are the parts that are low in stock (available quantity <3) and require reorder? And show all suppliers for that part

```
CREATE VIEW Low_Stock_View AS
```

```
SELECT availableparts.available_quantity, availableparts.part_id, vendor.vendor_name,  
vendor.vendor_phone
```

```
FROM availableparts
```

```
JOIN vendor ON availableparts.part_id = vendor.part_id
```

```
WHERE available_quantity < 3;
```

```
SELECT * from Low_Stock_View;
```