```
/* CH8 */
/* 27 */
CREATE PROCEDURE prc_cust_add(
IN CUST_NUM INT,
CUST_LNAME VARCHAR(50),
CUST_FNAME VARCHAR(50),
CUST_BALANCE DECIMAL(18,2))

INSERT INTO CH08_CUSTOMER (CUST_NUM, CUST_LNAME, CUST_FNAME, CUST_BALANCE)
VALUE (1002, 'Rauthor', 'Peter', 0.00);

/* 29 */

CREATE OR REPLACE TRIGGER trg_updatecustbalance2
AFTER INSERT ON LINE
FOR EACH ROW
BEGIN
UPDATE CUSTOMER
SET CUST_BALANCE = CUST_BALANCE - :OLD.INV_AMOUNT
WHERE CUST_NUM = :OLD.CUST_NUM
   END;

/* CH11 */

/* 1 */
/* EMP_SEX can have two values as the employee can be either male or female
as it can be either 'M' or 'F' nothing else, which means that we can call it low
sparsity
    */
/* 2 */
 /* EMP_SEX should not need to create the index because it indicates the low
sparsity
 as I menationed in question 1
 the requerment I think since it work same as any common indexes and this case it
need to have name indexes for composite index.alte
      create index EMP_IDX2 on
    EMPLOYEE (EMP_LNAME, EMP_FNAME);
 */

 /* 3 */
 /* in Excel */

 /* 22 */
 /*  the query given should show all of V_code, V_name, V_contract, V_state
with'212'
  so I would like to recommend to create index if equality comparison were there
 */

 /* 23 */
 /* the indexes should be created on V_STATE is show as:
 CREATE INDEX VDR_NDX2 is the name of the. index for V_ARECODE.
 as it should create index VDR_NDX2 on VENDOR (V_ARECODE);
 */

 /* 25 */
 /*  I think is one is highly recommened because it is preferred to create index
for the queries
 CREATE INDEX PROD_NDX1 ON
 PRODUCT(V_VODE);
```

```
 IN ORDER BY CLAUSE, P_CODE
 GROUP BY CALUSE
 */

 /* 26 */
 /*it should select statement is used to procure the data from the database. after
that it will procuring the data.
 from clause specifies one or more table from where records to be retieved
 and finally it will create index
 */

 /* 27 */
 select P_CODE, P_DESCRIPT, P_QOH, P_PRICE, V_CODE
 from PRODUCT
 WHERE P_REORDER = 50
 AND P_MIN = 50
 AND P_QOH<50
 ORDER BY P_QOH;

 /* 28 */
 /* I Would recommned
 create index prod_ndx1
 on PRODUCT (P_PREORDER); WHILE prod_ndx1 is P_MIN
 also
 create index prod_ndx3 on product(P_QOH);

 /* KCT */
 /* KCT */
/*
SQLyog Community Edition- MySQL GUI v8.12
MySQL - 5.1.40-community : Database - kct
*********************************************************************
*/
/*Table structure for table `KCT_costumetype` */

CREATE TABLE KCT_costumetype (
  TypeId SMALLINT(6) NOT NULL AUTO_INCREMENT,
  Description VARCHAR(35) NOT NULL,
  Photo VARCHAR(30) DEFAULT NULL,
  DailyRentalRate DECIMAL(8,2) NOT NULL,
  depositRate DECIMAL(8,2) DEFAULT NULL,
  replacementCost DECIMAL(8,2) DEFAULT NULL,
  PRIMARY KEY (TypeId)
) ;

/*Table structure for table `KCT_costume` */

CREATE TABLE KCT_costume (
  costumeID SMALLINT(6) NOT NULL,
  size VARCHAR(10) NOT NULL,
  available TINYINT(1) NOT NULL DEFAULT '1',
  datePurchased DATE DEFAULT NULL,
  costumeType SMALLINT(6) NOT NULL,
  PRIMARY KEY (costumeID),
  UNIQUE KEY costumeID (costumeID),
  KEY KCT_costume_ibfk_1 (costumeType),
  CONSTRAINT KCT_costume_ibfk_1 FOREIGN KEY (costumeType) REFERENCES
KCT_costumetype (TypeId)
) ;
```

```
/*Data for the table `costumetype` */

INSERT  INTO
KCT_costumetype(TypeId,Description,Photo,DailyRentalRate,depositRate,replacementCos
t)
VALUES (1,'Clown','clown.jpg','20.00','30.00','50.00'),
(2,'Ballerina','ballerina.jpg','20.00','20.00','50.00'),
(3,'Gorilla','gorilla.jpeg','30.00','23.00','50.00'),
(4,'Misc',NULL,'50.00',NULL,NULL),
(5,'star wars: Darth Vader','darth vader.jpg','40.00','30.00','70.00'),
(6,'star wars: luke','luke.jpg','40.00','30.00','70.00'),
(7,'star wars: leia','leia.jpg','40.00','30.00','70.00');


/*Data for the table `KCT_costume` */

INSERT  INTO KCT_costume(costumeID,size,available,datePurchased,costumeType)
VALUES (1,'medium',1,'2009-03-04',1),
(2,'large',1,'2009-04-04',3),
(3,'large',1,'2009-05-04',2),
(4,'small',0,'2009-06-04',2),
(5,'small',1,'2009-07-04',2),
(6,'small',0,'2009-07-14',5),
(7,'medium',1,'2009-08-14',7),
(8,'large',0,'2009-08-14',6),
(9,'medium',1,'2009-08-14',5),
(10,'medium',1,'2009-08-14',6),
(11,'large',1,'2009-02-04',2);

/*Table structure for table `KCT_customer` */

CREATE TABLE KCT_customer (
  CustomerID INT(11) NOT NULL,
  FirstName VARCHAR(25) NOT NULL,
  LastName VARCHAR(25) NOT NULL,
  Street VARCHAR(30) DEFAULT NULL,
  City VARCHAR(35) DEFAULT NULL,
  State CHAR(2) DEFAULT NULL,
  Zip VARCHAR(9) DEFAULT NULL,
  PhoneNumber CHAR(11) DEFAULT NULL,
  PRIMARY KEY (CustomerID)
) ;

/*Data for the table `KCT_customer` */

INSERT  INTO
KCT_customer(CustomerID,FirstName,LastName,Street,City,State,Zip,PhoneNumber)
VALUES (1,'Kalvin','Celcius','833 W 27th St.','Kearney','NE','68845','5559384768'),
(2,'Sue','Jones','433 W 27th St.','Kearney','NE','68845','5552342365'),
(3,'Carl','Carlson','431 Main St.','Grand Island','NE','68803','5553784832'),
(4,'Sally','Smith','123 A Ave','Aurora','NE','36257','5558434423'),
(5,'Dave','Johnsen','1208 Loveland Lane','Kearney','NE','68845','5552445623'),
(6,'John','Johnson','452 South Blvd','York','NE','36257','5558555366');

/*Table structure for table `KCT_rental` */

CREATE TABLE KCT_rental (
  RentalID INT(11) NOT NULL,
```

```
  RentalDate DATE NOT NULL,
  NumberOfDays DECIMAL(3,0) NOT NULL,
  TotalRentalAmount DECIMAL(8,2) NOT NULL,
  TotalDeposit DECIMAL(8,2) DEFAULT NULL,
  Tax DECIMAL(8,2) DEFAULT NULL,
  TotalAmount DECIMAL(8,2) NOT NULL,
  CustomerID INT(11) NOT NULL,
  ReturnDate DATE DEFAULT NULL,
  AmountReturned DECIMAL(8,2) DEFAULT NULL,
  PRIMARY KEY (RentalID),
  KEY RentalCustomerFK (CustomerID),
  CONSTRAINT RentalCustomerFK FOREIGN KEY (CustomerID) REFERENCES KCT_customer
(CustomerID) ON DELETE NO ACTION ON UPDATE NO ACTION
) ;

/*Data for the table `KCT_rental` */

INSERT  INTO
KCT_rental(RentalID,RentalDate,NumberOfDays,TotalRentalAmount,TotalDeposit,Tax,Tota
lAmount,CustomerID,ReturnDate,AmountReturned)
VALUES (1,'2009-10-02','4','80.00','30.00','7.70','87.70',1,NULL,NULL),
(2,'2009-07-08','7','40.00','20.00','9.75','49.75',2,'2009-07-15','15.00'),
(3,'2009-09-21','5','300.00','60.00','17.50','377.50',3,NULL,NULL),
(4,'2009-10-07','5','150.00','23.00','7.00','180.00',1,NULL,NULL),
(5,'2009-10-01','3','120.00','30.00','8.40','158.40',4,NULL,NULL),
(6,'2009-09-25','10','300.00','23.00','21.00','344.00',6,NULL,'100.00');

/*Table structure for table `KCT_lineitem` */

CREATE TABLE KCT_lineitem (
  RentalID INT(11) NOT NULL,
  CostumeID SMALLINT(6) NOT NULL,
  Rate DECIMAL(10,0) DEFAULT NULL,
  Deposit DECIMAL(10,0) DEFAULT NULL,
  PRIMARY KEY (RentalID,CostumeID),
  CONSTRAINT lineRentalFK FOREIGN KEY (RentalID) REFERENCES KCT_rental (RentalID),
  CONSTRAINT linecostumeFK FOREIGN KEY (CostumeID) REFERENCES KCT_costume
(costumeID)
) ;

/*Data for the table `KCT_lineitem` */

INSERT  INTO KCT_lineitem(RentalID,CostumeID,Rate,Deposit) VALUES
(1,1,'20','30'),
(2,3,'20','20'),
(3,4,'20','20'),
(3,6,'40','30'),
(4,2,'30','23'),
(5,8,'40','30'),
(6,2,'30','23');


CREATE VIEW KCT_customeraddress AS SELECT KCT_customer.FirstName AS
FirstName,KCT_customer.LastName AS LastName,
KCT_customer.Street AS Street,KCT_customer.City AS City,KCT_customer.State AS
State,KCT_customer.Zip AS Zip
FROM KCT_customer ;

CREATE VIEW KCT_laterentals AS SELECT KCT_customer.FirstName AS
```

```
FirstName,KCT_customer.LastName AS LastName,
KCT_customer.PhoneNumber AS PhoneNumber,KCT_rental.RentalDate AS RentalDate,
KCT_rental.TotalRentalAmount AS TotalRentalAmount,
KCT_rental.NumberOfDays AS NumberOfDays,(TO_DAYS(CURDATE()) -
TO_DAYS((KCT_rental.RentalDate + KCT_rental.NumberOfDays))) AS NumberOfDaysLate
FROM (KCT_customer JOIN KCT_rental)
WHERE ((KCT_customer.CustomerID = KCT_rental.CustomerID)
 AND (CURDATE() > (KCT_rental.RentalDate + INTERVAL KCT_rental.NumberOfDays DAY))
 AND ISNULL(KCT_rental.ReturnDate)) ;


/*View structure for view KCT_customerlatefees */

CREATE VIEW KCT_customerlatefees AS SELECT KCT_laterentals.LastName AS
LastName,KCT_laterentals.FirstName AS FirstName,
KCT_laterentals.PhoneNumber AS PhoneNumber,
(KCT_laterentals.NumberOfDaysLate * (KCT_laterentals.TotalRentalAmount /
KCT_laterentals.NumberOfDays)) AS LateFee
FROM KCT_laterentals ;

/*View structure for view KCT_customerlatetotals */

CREATE VIEW KCT_customerlatetotals AS SELECT KCT_customerlatefees.FirstName AS
FirstName,KCT_customerlatefees.LastName AS LastName,
KCT_customerlatefees.PhoneNumber AS PhoneNumber,SUM(KCT_customerlatefees.LateFee)
AS AccountDue
FROM KCT_customerlatefees
GROUP BY
KCT_customerlatefees.LastName,KCT_customerlatefees.FirstName,KCT_customerlatefees.P
honeNumber ;

/*View structure for view KCT_laterentals */
DROP PROCEDURE IF EXISTS KCT_NewCustomerWithRental$$
CREATE PROCEDURE KCT_NewCustomerWithRental(



)




create or replace TIGGER trg_KCT_updatecos
after delete on KCT_Costume
Begin
IF available is not 0
Delete From KCT_Costume
Where CostumeID = CostumeID
END if,
END;
Select * from KCT_Costume ;

CREATE TABLE KCT_Reservation(
CustomerID SMALLINT  NOT NULL,
CostumeID SMALLINT NOT NULL,
Re_Start_day Date,
Re_End_day Date,
PRIMARY KEY (CustomerID,CostumeID),
 CONSTRAINT lineRentalFK FOREIGN KEY (CustomerID) REFERENCES KCT_Customer
```

```
(CustomerID),
 CONSTRAINT linecostumeFK FOREIGN KEY (CostumeID) REFERENCES KCT_Costume
(costumeID));

create or replace procedure prc_KCT_Re_add3
after insert on KCT_Reservation
for each row
declare
begin
IF CustomerID is not null
IF CostumeID is not null
INSERT INTO KCT_Reservation
VALUES (FirtName, LastName, PhoneNumber, CostumeID,Re_Start_day,Re_End_day)
prc_cust_add('Jame', 'Trey','3023321900','2','2020-11-3','2020-11-29')
END;
SELECT * FROM CUSTOMER;
CALL KCT_Lineitem('Jame', 'Trey','3023321900','2','2020-11-3','2020-11-29',@msg);
SELECT @msg;

DROP PROCEDURE IF EXISTS KCT_NewCustomerWithRental$$
CREATE PROCEDURE KCT_NewCustomerWithRental(
IN inFirstName VARCHAR(25),
IN inLastName VARCHAR(25),
IN inPhoneNumber CHAR(11),
IN inNumberOfDays Decimal(8,2),
IN inCostumeID SMALLint (6)
OUT error_msg CHAR (80)
)

BEGIN

DECLARE rowCount SMALLINT,
DECLARE CID, RID INT DEFAULT 0,
DECLARE rateVAR DOUBLE,
DECLARE depositVAR DOUBLE;

      SET rowCount = 0;

SELECT count(*) INTO rowcount
FROM KCT_customer
WHERE KCT_customer.FirstName = inFirstName
      AND KCT_customer.LastName = inLastName
    AND KCT_customer.PhoneNumber = nPhoneNumber;

 IF rowcount > 0 THEN
      SET error_msg = 'Customer Already Exists -- No Action Taken',
 ELSE

    SELECT MAX(CustomerID) INTO CID
    FROM KCT_customer,

    SET CID = CID+1;

 INSERT INTO KCT_Customer
 (CustomerID, FirstName, LastName, PhoneNumber)
 VALUES (CID, inFirstName, inLastName, inPhoneNumber);

 SET error_msg = 'Customer Added';
```

```
 SELECT COUNT(*) INTO rowcount
 FROM KCT_customer C
 WHERE C.FirstName = inFirstName
 AND C.LastName = inLastName;

 IF rowcount = 1 THEN

 UPDATE KCT_costume
 SET available = 0
 WHERE costumeid = inCostumeID;

 SELECT MAX(RentalID) INTO RID
 FROM KCT_rental;

 SET RID = RID +1;
 SELECT DailyRentalRate INTO rateVAR
 FROM KCT_codtumetype
 WHERE typeid in
 (SELECT costumetype
 FROM KCT_costume
 WHERE costumeID = inCostumeID);

 SELECT DespositRate INTO depositVAR
 FROM KCT_codtumetype
 WHERE typeid in
 (SELECT costumetype
 FROM KCT_costume
 WHERE costumeID = inCostumeID);

 INSERT INTO KCT_lineitem(RentalID, CostumeID, rate, deposite)
 VALUES (RID, inCostumeID, rateVAR, depositVAR);

 INSERT INTO KCT_rental (RentalID, RentalDate, NumberOfDays, TotalRentalAmount,
TotalDeposit, Tax, TotalAmount, CustomerID)
 VALUES (RID,
 now(),
 NumberOfDays,
 1.07*(inNumberOfDays*rateVAR), depositVAR,
 .07*(inNumberOfDays*rateVAR),1.07*(inNumberOfDays*rateVAR) + depositVAR,
 CID);

 SET error_msg = 'Customer created, costume data update, Rental and line item
created';

            END IF,
            END IF,
 END$$

 DELIMITER ;


create or replace TIGGER trg_KCT_updatecos
after delete on KCT_Costume
Begin
IF available is not 0
Delete From KCT_Costume
Where CostumeID = CostumeID
END if,
END;
```

```
Select * from KCT_Costume ;

CREATE TABLE KCT_Reservation(
CustomerID SMALLINT  NOT NULL,
CostumeID SMALLINT NOT NULL,
Re_Start_day Date,
Re_End_day Date,
PRIMARY KEY (CustomerID,CostumeID),
 CONSTRAINT lineRentalFK FOREIGN KEY (CustomerID) REFERENCES KCT_Customer
(CustomerID),
 CONSTRAINT linecostumeFK FOREIGN KEY (CostumeID) REFERENCES KCT_Costume
(costumeID));

create or replace procedure prc_KCT_Re_add3
after insert on KCT_Reservation
for each row
declare
begin
IF CustomerID is not null
IF CostumeID is not null
INSERT INTO KCT_Reservation
VALUES (FirtName, LastName, PhoneNumber, CostumeID,Re_Start_day,Re_End_day)
prc_cust_add('Jame', 'Trey','3023321900','2','2020-11-3','2020-11-29')
END;
SELECT * FROM CUSTOMER;
CALL KCT_Lineitem('Jame', 'Trey','3023321900','2','2020-11-3','2020-11-29',@msg);
SELECT @msg;
```