**Problem Sheet – VII (MADHUMITHA.S – 19PW13)**

CREATE TABLE IF NOT EXISTS COURSES  
(  
 CCODE VARCHAR(10),  
 CNAME VARCHAR(50) NOT NULL ,  
 COURSEFEE NUMERIC(6 ,2) NOT NULL ,  
 DURATION NUMERIC(3) NOT NULL ,  
 PREREQ VARCHAR(100),  
 CONSTRAINT ccode\_pk PRIMARY KEY (CCODE)  
);  
  
CREATE TABLE IF NOT EXISTS BATCHES  
(  
 BATCHCODE VARCHAR(12) CHECK ( BATCHCODE REGEXP ('^[A-Z]{2,6}[0-9]{6}$')),  
 CCODE VARCHAR(10),  
 STDATE DATE ,  
 ENDDATE DATE ,  
 TIMINGS VARCHAR(20),  
 CONSTRAINT ccode\_fk FOREIGN KEY (CCODE) REFERENCES COURSES(CCODE),  
 CONSTRAINT bcode\_pk PRIMARY KEY (BATCHCODE),  
 CONSTRAINT endtime\_check CHECK ( STDATE < ENDDATE )  
);  
  
CREATE TABLE IF NOT EXISTS STUDENTS  
(  
 ADMNO NUMERIC(5),  
 BATCHCODE VARCHAR(12),  
 ROLLNO NUMERIC(3) ,  
 FULLNAME VARCHAR(50),  
 FATHERNAME VARCHAR(50),  
 EMAIL VARCHAR(50) UNIQUE,  
 PHONENO VARCHAR(20) UNIQUE CHECK ( PHONENO REGEXP ('(\+[0-9]{2}-[0-9]{10})|[0-9]{10}')) ,  
 DJ DATE,  
 CONSTRAINT bcode\_fk FOREIGN KEY (BATCHCODE) REFERENCES BATCHES(BATCHCODE),  
 CONSTRAINT admno\_pk PRIMARY KEY (ADMNO)  
);  
  
CREATE TABLE IF NOT EXISTS PAYMENTS  
(  
 RCPTNO NUMERIC(5),  
 ADMNO NUMERIC(5),  
 AMOUNT NUMERIC(6,2),  
 PAYDATE DATE,  
 REMARKS VARCHAR(200),  
 CONSTRAINT rcptno\_pk PRIMARY KEY (RCPTNO),  
 CONSTRAINT admno\_fk FOREIGN KEY (ADMNO) REFERENCES STUDENTS(ADMNO)  
);

1. Insert the following data into these tables.

# Courses

INSERT INTO COURSES VALUES(’ORACLE11G’,’ORACLE DATABASE 11G’,2500,40,’COMPUTER KONWLEDGE’); INSERT INTO COURSES VALUES(’JAVASE6.0’,’JAVA SE 6.0’,2500,40,’C LANGUAGE’);

INSERT INTO COURSES VALUES(’DOTNET3.5’,’MICROSOFT .NET 3.5’,3750,80,’C LANGUAGE & SQL’);

INSERT INTO COURSES VALUES  
('ORACLE11G','ORACLE DATABASE 11G',2500,40,'COMPUTER KONWLEDGE'),  
('JAVASE6.0','JAVA SE 6.0',2500,40,'C LANGUAGE'),  
('DOTNET3.5','MICROSOFT.NET 3.5',3750,80,'C LANGUAGE & SQL');

Batches

INSERT INTO BATCHES VALUES(’ORA130508’,’ORACLE11G’,’13-MAY-08’,’17-JUN-08’,’4:30 TO 6:00 PM’); INSERT INTO BATCHES VALUES(’DOTNET130508’,’DOTNET3.5’,’13-MAY-08’,’26-JUN-08’,’7:00 TO 9:00 AM’); INSERT INTO BATCHES VALUES(’ORA270608’,’ORACLE11G’,’27-JUN-08’,NULL,’5:00 TO 6:00 PM’);

INSERT INTO BATCHES VALUES(’JS270608’,’JAVASE6.0’,’27-JUN-08’,NULL,’6:00 TO 7:00 PM’);

INSERT INTO BATCHES VALUES  
('ORA130508','ORACLE11G','08-05-13','17-06-08','4:30 TO 6:00 PM'),  
('DOTNET130508','DOTNET3.5','08-05-13','08-06-26','7:00 TO 9:00 AM'),  
('ORA270608','ORACLE11G','08-06-27',NULL,'5:00 TO 6:00 PM'),  
('JS270608','JAVASE6.0','08-06-27',NULL,'6:00 TO 7:00 PM');

Students

INSERT INTO STUDENTS VALUES(1,’ORA130508’,1,’MICHEAL JORDON’, ’TIM JORDON’,’[MJORDON@YAHOO.COM](mailto:MJORDON@YAHOO.COM)’, ’9873737334’,’11-MAY-08’);

INSERT INTO STUDENTS VALUES(2,’ORA130508’,2,’TIM SLIM’, ’TIM KEN’,’[TIM@YAHOO.COM](mailto:TIM@YAHOO.COM)’, ’9833334334’,’11-MAY-08’);

INSERT INTO STUDENTS VALUES(3,’DOTNET130508’,1,’HUNTER JASON’, ’HUNTER BOB’,’[JHUNTER@YAHOO.COM](mailto:JHUNTER@YAHOO.COM)’, ’34344343’,’11-MAY-08’);

INSERT INTO STUDENTS VALUES(4,’JS270608’,1,’JAMES GOODWILL’, ’JAMES ROBERTS’,’[JAMES@YAHOO.COM](mailto:JAMES@YAHOO.COM)’, ’9989898998’,’26-JUN-08’);

INSERT INTO STUDENTS VALUES(5,’JS270608’,2,’KENNY PETERSON’, ’KENNY JACOB’,’[KPERERSON@GMAIL.COM](mailto:KPERERSON@GMAIL.COM)’, ’9983373333’,’27-JUN-08’);

INSERT INTO STUDENTS VALUES(6,’ORA270608’,1,’GLEN JHONSON’, ’GLEN HENDRICK’,’[GLEN@GMAIL.COM](mailto:GLEN@GMAIL.COM)’, ’9898398985’,’28-JUN-08’);

INSERT INTO STUDENTS VALUES(7,’ORA270608’,2,’BATES KATHY’, ’BATES ROBERTS’,’[KATHY@YMAIL.COM](mailto:KATHY@YMAIL.COM)’, ’234423232’,’30-JUN-08’);

INSERT INTO STUDENTS VALUES  
(1,'ORA130508',1,'MICHEAL JORDON', 'TIM JORDON','MJORDON@YAHOO.COM', '9873737334','08-05-11'),  
(2,'ORA130508',2,'TIM SLIM', 'TIM KEN','TIM@YAHOO.COM', '9833334334','08-05-11'),  
(3,'DOTNET130508',1,'HUNTER JASON', 'HUNTER BOB','JHUNTER@YAHOO.COM', '34344343','08-05-11'),  
(4,'JS270608',1,'JAMES GOODWILL', 'JAMES ROBERTS','JAMES@YAHOO.COM', '9989898998','08-06-26'),  
(5,'JS270608',2,'KENNY PETERSON', 'KENNY JACOB','KPERERSON@GMAIL.COM', '9983373333','08-06-27'),  
(6,'ORA270608',1,'GLEN JHONSON', 'GLEN HENDRICK','GLEN@GMAIL.COM', '9898398985','08-06-28'),  
(7,'ORA270608',2,'BATES KATHY', 'BATES ROBERTS','KATHY@YMAIL.COM', '234423232','08-06-30');

# Payments

INSERT INTO PAYMENTS VALUES(1,1,300,’11-MAY-08’,’REG. FEE’);

INSERT INTO PAYMENTS VALUES(2,2,2500,’11-MAY-08’,’TOTAL FEE’);

INSERT INTO PAYMENTS VALUES(3,3,1000,’11-MAY-08’,’REG. FEE’);

INSERT INTO PAYMENTS VALUES(4,3,2750,’12-MAY-08’,NULL);

INSERT INTO PAYMENTS VALUES(5,4,300,’26-JUN-08’,’REG. FEE’);

INSERT INTO PAYMENTS VALUES(6,5,300,’27-JUN-08’,’REG. FEE’);

INSERT INTO PAYMENTS VALUES(7,4,1700,’27-JUN-08’,NULL);

INSERT INTO PAYMENTS VALUES(8,5,1700,’29-JUN-08’,NULL);

INSERT INTO PAYMENTS VALUES(9,6,2500,’28-JUN-08’,’CHEQUE NO:3434343 SBI DWK’);

INSERT INTO PAYMENTS VALUES(10,7,2500,’30-JUN-08’,NULL);

INSERT INTO PAYMENTS VALUES  
(1,1,300,'08-05-11','REG. FEE'),  
(2,2,2500,'08-05-11','TOTAL FEE'),  
(3,3,1000,'08-05-11','REG. FEE'),  
(4,3,2750,'08-05-12',NULL),  
(5,4,300,'08-06-26','REG. FEE'),  
(6,5,300,'08-06-27','REG. FEE'),  
(7,4,1700,'08-06-27',NULL),  
(8,5,1700,'08-06-29',NULL),  
(9,6,2500,'08-06-28','CHEQUE NO:3434343 SBI DWK'),  
(10,7,2500,'08-06-30',NULL);

Simple Queries

1. Display the details of all students in the ascending order of batchcode and joining date

SELECT \*

FROM STUDENTS

ORDER BY BATCHCODE, DJ;

1. Display all payments made in the month of May, 2008

SELECT \*

FROM PAYMENTS

WHERE YEAR(PAYDATE) = 2008

AND MONTH(PAYDATE) = 5;

1. Display all payment made through cheque.

SELECT \*

FROM PAYMENTS

WHERE REMARKS LIKE 'CHEQUE%';

1. Display student name, fathername, joining date and number of days since joined.

SELECT FULLNAME, FATHERNAME, DJ, DAY(CURRENT\_DATE - DJ)

FROM STUDENTS;

1. Display the batches that are currently running.

SELECT \*

FROM BATCHES

WHERE ENDDATE IS NULL

OR ENDDATE > CURRENT\_DATE;

1. Display the batches of JAVASE and ORACLE.

SELECT \*

FROM BATCHES

WHERE CCODE LIKE '%ORACLE%'

OR '%JAVASE%';

1. Display the due date for the payment assuming due date is 7 days from dj.

SELECT \*, TIMESTAMPADD(DAY, 7, DJ) AS 'Last DAY TO PAY'

FROM STUDENTS;

1. Display the details of students where due date for payment is over.

SELECT \*

FROM STUDENTS

WHERE TIMESTAMPADD(DAY, 7, DJ) < CURRENT\_DATE;

1. Display the details of courses with a proposed increase of 10% in course fee for courses with course fee less than 3000.

SELECT CCODE, CNAME, DURATION, PREREQ, COURSEFEE \* 1.1 AS 'COURSEFEE'

FROM COURSES

WHERE COURSEFEE < 3000;

1. Display the students whose name contains letter 's' and father's name contains letter 'p'.

SELECT \*

FROM STUDENTS

WHERE FULLNAME LIKE '%s%'

AND FATHERNAME LIKE '%p%';

1. Display the batches that are running for more than 45 days.

SELECT \*

FROM BATCHES

WHERE ENDDATE IS NULL

OR DAY(TIMEDIFF(ENDDATE, STDATE)) > 45;

1. Display batchcode, stdate and approximate ending date for Oracle batches that are currently running, if batch takes two months.

SELECT BATCHCODE, STDATE, TIMESTAMPADD(MONTH, 2, STDATE) AS 'ESTIMATED ENDING DATE'

FROM BATCHES

WHERE CCODE LIKE 'ORACLE%';

1. Display the difference between actual ending date and estimated ending date for Java bathes assuming each batch takes two months.

SELECT BATCHCODE,

STDATE,

TIMESTAMPADD(MONTH, 2, STDATE) AS 'ESTIMATED ENDING DATE',

ENDDATE - TIMESTAMPADD(MONTH, 2, STDATE)

FROM BATCHES

WHERE CCODE LIKE 'JAVA%'

AND ENDDATE IS NOT NULL;

1. Insert course details of JAVA EE web course.

INSERT INTO COURSES VALUES ('JAVAEEWEB','JAVA EE (WEB APPLICATIONS)', 3000, 40,'JAVA LANG AND SQL');

1. Update the batches table to set enddate of batch JS130508 to yesterday.

UPDATE BATCHES

SET ENDDATE = TIMESTAMPDIFF(DAY, 1, CURRENT\_DATE)

WHERE BATCHCODE = 'JS130508';

1. Display the batches that started in the previous year but ended in this year.

SELECT \*

FROM BATCHES

WHERE YEAR(STDATE) = YEAR(ENDDATE) - 1

AND YEAR(ENDDATE) = YEAR(CURRENT\_DATE);

1. Display the payments with amount more than 1000 or made by students with ADMNO in the range 100 and 150 in the last 10 days.

SELECT \*

FROM PAYMENTS

WHERE AMOUNT > 1000

OR (ADMNO BETWEEN 100 AND 150 AND

DATEDIFF(PAYDATE, CURRENT\_DATE) < 10);

1. Change the paydate for receipt 12 to 1st June, 2008 and admno to 120.

UPDATE PAYMENTS

SET ADMNO = 120 AND PAYDATE = '01-06-08'

WHERE RCPTNO = 12;

1. Display student's name, batchcode, and DJ in ascending order or name followed by DJ.

SELECT FULLNAME, BATCHCODE, DJ

FROM STUDENTS

ORDER BY FULLNAME, DJ;

1. Display the approximate date when cheque will be realized for cheque payments.

Grouping

1. Display the total amount paid by all students.

SELECT SUM(AMOUNT) AS "TOTAL AMOUNT PAID"

FROM PAYMENTS

GROUP BY ADMNO;

1. Display the highest receipt number for payments in the month of May 2008.

SELECT MAX(RCPTNO)

FROM PAYMENTS

WHERE YEAR(PAYDATE) = 2008

AND MONTH(PAYDATE) = 5

GROUP BY YEAR(PAYDATE), MONTH(PAYDATE);

1. Display the batchcode and the number of students in the batch.

SELECT BATCHCODE, COUNT(ADMNO)

FROM STUDENTS

GROUP BY BATCHCODE;

1. Display the most recently stated batched for each course.

SELECT CCODE, MAX(STDATE) AS "MOST RECENTLY STARTED"

FROM BATCHES

GROUP BY CCODE;

1. Display the total amount paid on each day.

SELECT SUM(AMOUNT) AS PAYMENT

FROM PAYMENTS

GROUP BY PAYDATE;

1. Display the number of batches for each course in the current year.

SELECT CCODE, COUNT(BATCHCODE) AS "BATCH COUNT"

FROM BATCHES

GROUP BY CCODE;

1. Display the amount collected for each month.

SELECT MONTHNAME(PAYDATE) AS MONTH, YEAR(PAYDATE), SUM(AMOUNT) AS "AMOUNT COLLECTED"

FROM PAYMENTS

GROUP BY MONTHNAME(PAYDATE), YEAR(PAYDATE);

1. Display the batches where the number of students is more than 10.

SELECT BATCHCODE

FROM STUDENTS

GROUP BY BATCHCODE

HAVING COUNT(BATCHCODE) > 10;

1. Display the batchcode and first and last admission into batch.

SELECT BATCHCODE, MIN(ADMNO), MAX(ADMNO)

FROM STUDENTS

GROUP BY BATCHCODE;

1. Display the courses for more than a batch was started in the same month.

SELECT CCODE

FROM BATCHES

GROUP BY CCODE, MONTH(STDATE), YEAR(STDATE)

HAVING COUNT(BATCHCODE) > 1;

1. Display the number of batches for each year and course.

SELECT CCODE, YEAR(STDATE), COUNT(BATCHCODE)

FROM BATCHES

GROUP BY CCODE, YEAR(STDATE);

1. Display how many batches are currently running.

SELECT COUNT(\*)

FROM BATCHES

WHERE ENDDATE IS NULL

OR ENDDATE > CURRENT\_DATE;

1. Display the number of students using each mail server.

SELECT SUBSTRING(EMAIL, LOCATE('@', EMAIL) + 1, LENGTH(EMAIL) - 1) AS SERVER, COUNT(ADMNO)

FROM STUDENTS

GROUP BY SERVER;

1. Display the number of students for each batch on java with more than 10 students in the ascending order of no. Of students.

SELECT COUNT(ADMNO)

FROM STUDENTS

GROUP BY BATCHCODE

HAVING COUNT(BATCHCODE) > 10

ORDER BY COUNT(BATCHCODE);

1. Display the total amount received for the current month.

SELECT MONTH(CURRENT\_DATE), SUM(AMOUNT)

FROM PAYMENTS

WHERE MONTH(PAYDATE) = MONTH(CURRENT\_DATE)

GROUP BY MONTH(PAYDATE);

1. Display the year and total payments for the year.

SELECT YEAR(PAYDATE), SUM(AMOUNT) AS 'TOTAL PAYMENTS'

FROM PAYMENTS

GROUP BY YEAR(PAYDATE);

1. Display the number of courses students where name contains 'tom' have done.

SELECT S.FULLNAME,COUNT(B.CCODE) AS COURSE\_COUNT

FROM STUDENTS S

INNER JOIN BATCHES B on S.BATCHCODE = B.BATCHCODE

WHERE S.FULLNAME LIKE '%TIM%'

GROUP BY S.FULLNAME;

1. Display the days on which more than 5000 was received as payments.

SELECT PAYDATE

FROM PAYMENTS

GROUP BY PAYDATE

HAVING SUM(AMOUNT) > 5000;

1. Display the batches where number of students who joined after in the last 10 days are more than 5.

SELECT BATCHCODE

FROM STUDENTS

WHERE TIMESTAMPDIFF(DAY, 10, CURRENT\_DATE) <= DJ

AND DJ < CURRENT\_DATE

GROUP BY BATCHCODE

HAVING COUNT(BATCHCODE) > 5;

Joining

1. Display the batchcode, course name, starting date.

SELECT BATCHCODE, CCODE, STDATE

FROM BATCHES;

1. Display rcptno, fullname, amount paid and pay date in the order of paydate.

SELECT RCPTNO, FULLNAME, AMOUNT, PAYDATE

FROM PAYMENTS P

INNER JOIN STUDENTS S USING (ADMNO)

ORDER BY PAYDATE;

1. Display course name, batchcode and fullname.

SELECT CCODE, BATCHCODE, FULLNAME

FROM BATCHES

INNER JOIN STUDENTS USING (BATCHCODE);

1. Display the number of students joined for each course.

SELECT CCODE, COUNT(CCODE)

FROM BATCHES

INNER JOIN STUDENTS S USING (BATCHCODE)

GROUP BY CCODE;

1. Display the amount paid by each student in batch 'ora130508'.

SELECT ADMNO, SUM(AMOUNT)

FROM STUDENTS

INNER JOIN PAYMENTS USING (ADMNO)

WHERE BATCHCODE = 'ora130508'

GROUP BY ADMNO;

1. Display the amount paid by each student in batch 'ora130508'.

SELECT ADMNO, SUM(AMOUNT)

FROM STUDENTS

INNER JOIN PAYMENTS USING (ADMNO)

WHERE BATCHCODE = 'ora130508'

GROUP BY ADMNO;

1. Display the details of batches for course with course fee more than 3000.

SELECT B.\*

FROM COURSES

INNER JOIN BATCHES B USING (CCODE)

WHERE COURSEFEE > 3000;

1. Display the rcptno, fullname, batchcode, amount, paydate for payments in the last 10 days.

SELECT RCPTNO, FULLNAME, BATCHCODE, AMOUNT, PAYDATE

FROM STUDENTS

INNER JOIN PAYMENTS USING (ADMNO)

WHERE DATEDIFF(CURRENT\_DATE, PAYDATE) < 10;

1. Display the cname, batchcode, stdate and enddate for all batches that are completed.

SELECT CNAME, BATCHCODE, STDATE, ENDDATE

FROM BATCHES

INNER JOIN COURSES USING (CCODE)

WHERE ENDDATE IS NOT NULL

OR ENDDATE < CURRENT\_DATE;

1. Display the fullname, dj and amountpaid at the time of joining.

SELECT FULLNAME, DJ, AMOUNT

FROM STUDENTS

INNER JOIN PAYMENTS USING (ADMNO)

WHERE DJ = PAYDATE;

1. Display the coursename, batchcode including courses that do not have any batches.

SELECT CNAME, BATCHCODE

FROM BATCHES

RIGHT JOIN COURSES USING (CCODE);

1. Display the names of the students who have not paid anything so far.

SELECT S.\*

FROM STUDENTS S

LEFT JOIN PAYMENTS USING (ADMNO)

WHERE AMOUNT IS NULL;

1. Display the batches that started after batch with code 'ora130508'.

SELECT B.\*

FROM BATCHES A

INNER JOIN BATCHES B on A.STDATE < B.STDATE

WHERE A.BATCHCODE = 'ora130508';

1. Display the fullname and the batchcode for students who have paid total amount at the time of admission.

SELECT FULLNAME, BATCHCODE

FROM STUDENTS

INNER JOIN BATCHES USING (BATCHCODE)

INNER JOIN COURSES USING (CCODE)

INNER JOIN (SELECT ADMNO, SUM(AMOUNT) AS 'AMOUNT' FROM PAYMENTS GROUP BY ADMNO) P USING (ADMNO)

WHERE AMOUNT = COURSEFEE;

1. Display the details of students who have dues.

SELECT S.\*

FROM STUDENTS S

INNER JOIN BATCHES USING (BATCHCODE)

INNER JOIN COURSES USING (CCODE)

INNER JOIN (SELECT ADMNO, SUM(AMOUNT) AS 'AMOUNT' FROM PAYMENTS GROUP BY ADMNO) P USING (ADMNO)

WHERE AMOUNT < COURSEFEE;

Subqueries

1. Display the payments made by student 'JAMES GOODWILL'.

SELECT \*

FROM PAYMENTS

WHERE ADMNO = (SELECT ADMNO FROM STUDENTS WHERE FULLNAME = 'JAMES GOODWILL');

1. Display the payments made by students who joined into 'ORA270608' batch.

SELECT \*

FROM PAYMENTS

WHERE ADMNO IN (SELECT ADMNO FROM STUDENTS WHERE BATCHCODE = 'ORA270608');

1. Display the batches for course with duration more than 40 hours.

SELECT BATCHCODE

FROM BATCHES

WHERE CCODE IN (SELECT CCODE FROM COURSES WHERE DURATION > 40);

1. Display the students who made payments in the current month.

SELECT FULLNAME

FROM STUDENTS

WHERE ADMNO IN (SELECT DISTINCT ADMNO FROM PAYMENTS WHERE MONTH(PAYDATE) = MONTH(CURRENT\_DATE));

1. Display the students who joined into oracle course.

SELECT FULLNAME

FROM STUDENTS

WHERE BATCHCODE IN (SELECT BATCHCODE FROM BATCHES WHERE CCODE LIKE '%ORACLE%');

1. Display the details of students from currently running batches.

SELECT \*

FROM STUDENTS

WHERE BATCHCODE IN (SELECT BATCHCODE FROM BATCHES WHERE ENDDATE IS NULL OR CURRENT\_DATE < ENDDATE);

1. Display the fullname, email address of all students who completed batch 6 months back.

SELECT FULLNAME, EMAIL

FROM STUDENTS

WHERE BATCHCODE IN (SELECT BATCHCODE FROM BATCHES WHERE MONTH(ENDDATE) = MONTH(TIMESTAMPDIFF(MONTH, 6, CURRENT\_DATE)));

1. Display the fullname, email address of .net students who completed batch 6 months back.

SELECT FULLNAME, EMAIL

FROM STUDENTS

WHERE BATCHCODE IN (SELECT BATCHCODE

FROM BATCHES

WHERE MONTH(ENDDATE) = MONTH(TIMESTAMPDIFF(MONTH, 6, CURRENT\_DATE))

AND CCODE LIKE '%DOTNET%');

1. Display the details of students who belonged to batch with less than 10 students.

SELECT BATCHCODE, COUNT(BATCHCODE) AS COUNT

FROM STUDENTS

GROUP BY BATCHCODE

HAVING COUNT(BATCHCODE) < 10;

1. Display the rcptno, fullname, batchcode, amount, and paydate for students who joined into Oracle course.

SELECT RCPTNO, FULLNAME, BATCHCODE, AMOUNT, PAYDATE

FROM STUDENTS

INNER JOIN PAYMENTS USING (ADMNO)

WHERE BATCHCODE IN (

SELECT BATCHCODE

FROM BATCHES

WHERE CCODE LIKE '%ORACLE%');

1. Display the course for which we have any batch with more than 10 students.

SELECT DISTINCT CCODE

FROM BATCHES

WHERE BATCHCODE IN (SELECT BATCHCODE FROM STUDENTS GROUP BY BATCHCODE HAVING COUNT(\*) > 10);

1. Display the batches for which the total amount collected is more than 20000.

SELECT BATCHCODE, SUM(AMOUNT) AS COLLECTED\_AMOUNT

FROM STUDENTS

INNER JOIN PAYMENTS USING (ADMNO)

GROUP BY BATCHCODE

HAVING SUM(AMOUNT) > 20000;

1. Display the students who did not join on the date of starting of the batch.

SELECT FULLNAME

FROM STUDENTS

WHERE ADMNO NOT IN (SELECT ADMNO FROM STUDENTS WHERE DJ IN (SELECT STDATE FROM BATCHES));

1. Display the batches with top 3 highest no. Of students.

SELECT BATCHCODE

FROM BATCHES

WHERE BATCHCODE IN (

SELECT DISTINCT BATCHCODE

FROM STUDENTS

GROUP BY BATCHCODE

ORDER BY COUNT(ADMNO) DESC)

LIMIT 3;

1. Update the amount in receipt 200 with total amount for course in which student joined.

UPDATE PAYMENTS

SET AMOUNT =(

SELECT \*

FROM (SELECT COURSEFEE

FROM COURSES

WHERE CCODE = (SELECT CCODE

FROM BATCHES

WHERE BATCHCODE = (SELECT BATCHCODE

FROM STUDENTS

WHERE ADMNO = (SELECT ADMNO FROM PAYMENTS WHERE RCPTNO = '200')))) t)

WHERE RCPTNO = 200;