use 19pw12;  
  
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,  
 DJ DATE,  
 CONSTRAINT batchcode\_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)  
);  
  
# Inserting Courses records  
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 &amp; SQL');  
  
# Inserting Batches records  
INSERT INTO BATCHES  
VALUES ('ORA130508', 'ORACLE11G', STR\_TO\_DATE('13-MAY-08', '%d-%b-%y'), STR\_TO\_DATE('17-JUN-08', '%d-%b-%y'),  
 '4:30 TO 6:00 PM'),  
 ('DOTNET130508', 'DOTNET3.5', STR\_TO\_DATE('13-MAY-08', '%d-%b-%y'), STR\_TO\_DATE('26-JUN-08', '%d-%b-%y'),  
 '7:00 TO 9:00 AM'),  
 ('ORA270608', 'ORACLE11G', STR\_TO\_DATE('27-JUN-08', '%d-%b-%y'), NULL, '5:00 TO 6:00 PM'),  
 ('JS270608', 'JAVASE6.0', STR\_TO\_DATE('27-JUN-08', '%d-%b-%y'), NULL, '6:00 TO 7:00 PM');  
  
# Inserting Students records  
INSERT INTO STUDENTS  
VALUES (1, 'ORA130508', 1, 'MICHEAL JORDON', 'TIM JORDON', 'MJORDON@YAHOO.COM', '9873737334',  
 STR\_TO\_DATE('11-MAY-08', '%d-%b-%y')),  
 (2, 'ORA130508', 2, 'TIM SLIM', 'TIM KEN', 'TIM@YAHOO.COM', '9833334334', STR\_TO\_DATE('11-MAY-08', '%d-%b-%y')),  
 (3, 'DOTNET130508', 1, 'HUNTER JASON', 'HUNTER BOB', 'JHUNTER@YAHOO.COM', '34344343',  
 STR\_TO\_DATE('11-MAY-08', '%d-%b-%y')),  
 (4, 'JS270608', 1, 'JAMES GOODWILL', 'JAMES ROBERTS', 'JAMES@YAHOO.COM', '9989898998',  
 STR\_TO\_DATE('26-JUN-08', '%d-%b-%y')),  
 (5, 'JS270608', 2, 'KENNY PETERSON', 'KENNY JACOB', 'KPERERSON@GMAIL.COM', '9983373333',  
 STR\_TO\_DATE('27-JUN-08', '%d-%b-%y')),  
 (6, 'ORA270608', 1, 'GLEN JHONSON', 'GLEN HENDRICK', 'GLEN@GMAIL.COM', '9898398985',  
 STR\_TO\_DATE('28-JUN-08', '%d-%b-%y')),  
 (7, 'ORA270608', 2, 'BATES KATHY', 'BATES ROBERTS', 'KATHY@YMAIL.COM', '234423232',  
 STR\_TO\_DATE('30-JUN-08', '%d-%b-%y'));  
  
# Inserting Payments records  
INSERT INTO PAYMENTS  
VALUES (1, 1, 300, STR\_TO\_DATE('11-MAY-08', '%d-%b-%y'), 'REG. FEE'),  
 (2, 2, 2500, STR\_TO\_DATE('11-MAY-08', '%d-%b-%y'), 'TOTAL FEE'),  
 (3, 3, 1000, STR\_TO\_DATE('11-MAY-08', '%d-%b-%y'), 'REG. FEE'),  
 (4, 3, 2750, STR\_TO\_DATE('12-MAY-08', '%d-%b-%y'), NULL),  
 (5, 4, 300, STR\_TO\_DATE('26-JUN-08', '%d-%b-%y'), 'REG. FEE'),  
 (6, 5, 300, STR\_TO\_DATE('27-JUN-08', '%d-%b-%y'), 'REG. FEE'),  
 (7, 4, 1700, STR\_TO\_DATE('27-JUN-08', '%d-%b-%y'), NULL),  
 (8, 5, 1700, STR\_TO\_DATE('29-JUN-08', '%d-%b-%y'), NULL),  
 (9, 6, 2500, STR\_TO\_DATE('28-JUN-08', '%d-%b-%y'), 'CHEQUE NO:3434343SBI DWK'),  
 (10, 7, 2500, STR\_TO\_DATE('30-JUN-08', '%d-%b-%y'), NULL);  
  
  
  
## SIMPLE QUERIES  
  
# Ques no-01  
SELECT \*  
FROM STUDENTS  
ORDER BY BATCHCODE, DJ;  
  
# Ques no-02  
SELECT \*  
FROM PAYMENTS  
WHERE YEAR(PAYDATE) = 2008  
 AND MONTH(PAYDATE) = 5;  
  
# Ques no-03  
SELECT \*  
FROM PAYMENTS  
WHERE REMARKS LIKE 'CHEQUE%';  
  
# Ques no-04  
SELECT FULLNAME, FATHERNAME, DJ, DAY(CURRENT\_DATE - DJ)  
FROM STUDENTS;  
  
# Ques no-05  
SELECT \*  
FROM BATCHES  
WHERE ENDDATE IS NULL  
 OR ENDDATE > CURRENT\_DATE;  
  
# Ques no-06  
SELECT \*  
FROM BATCHES  
WHERE CCODE LIKE '%ORACLE%'  
 OR '%JAVASE%';  
  
# Ques no-07  
SELECT \*, TIMESTAMPADD(DAY, 7, DJ) AS 'Last DAY TO PAY'  
FROM STUDENTS;  
  
# Ques no-08  
SELECT \*  
FROM STUDENTS  
WHERE TIMESTAMPADD(DAY, 7, DJ) < CURRENT\_DATE;  
  
# Ques no-09  
SELECT CCODE, CNAME, DURATION, PREREQ, COURSEFEE \* 1.1 AS 'COURSEFEE'  
FROM COURSES  
WHERE COURSEFEE < 3000;  
  
# Ques no-10  
SELECT \*  
FROM STUDENTS  
WHERE FULLNAME LIKE '%s%'  
 AND FATHERNAME LIKE '%p%';  
  
# Ques no-11  
SELECT \*  
FROM BATCHES  
WHERE ENDDATE IS NULL  
 OR DAY(TIMEDIFF(ENDDATE, STDATE)) > 45;  
  
# Ques no-12  
SELECT BATCHCODE, STDATE, TIMESTAMPADD(MONTH, 2, STDATE) AS 'ESTIMATED ENDING DATE'  
FROM BATCHES  
WHERE CCODE LIKE 'ORACLE%';  
  
# Ques no-13  
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;  
  
# Ques no-14  
INSERT INTO COURSES VALUE ('JAVAEEWEB', 'JAVA EE(WEB APPLICATIONS)', 3000, 40, 'JAVA LANG AND SQL');  
  
# Ques no-15  
UPDATE BATCHES  
SET ENDDATE = TIMESTAMPDIFF(DAY, 1, CURRENT\_DATE)  
WHERE BATCHCODE = 'JS130508';  
  
# Ques no-16  
SELECT \*  
FROM BATCHES  
WHERE YEAR(STDATE) = YEAR(ENDDATE) - 1  
 AND YEAR(ENDDATE) = YEAR(CURRENT\_DATE);  
  
# Ques no-17  
SELECT \*  
FROM PAYMENTS  
WHERE AMOUNT > 1000  
 OR (ADMNO BETWEEN 100 AND 150 AND  
 DATEDIFF(PAYDATE, CURRENT\_DATE) < 10);  
  
# Ques no-18  
UPDATE PAYMENTS  
SET ADMNO = 120 AND PAYDATE = '01-06-08'  
WHERE RCPTNO = 12;  
  
# Ques no-19  
SELECT FULLNAME, BATCHCODE, DJ  
FROM STUDENTS  
ORDER BY FULLNAME, DJ;  
  
  
  
  
# Grouping  
  
# Ques no Grp-01  
SELECT SUM(AMOUNT) AS "TOTAL AMOUNT PAID"  
FROM PAYMENTS  
GROUP BY ADMNO;  
  
# Ques no Grp-02  
SELECT MAX(RCPTNO)  
FROM PAYMENTS  
WHERE YEAR(PAYDATE) = 2008  
 AND MONTH(PAYDATE) = 5  
GROUP BY YEAR(PAYDATE), MONTH(PAYDATE);  
# Do we need group by?  
  
# Ques no Grp-03  
SELECT BATCHCODE, COUNT(ADMNO)  
FROM STUDENTS  
GROUP BY BATCHCODE;  
  
# Ques no Grp-04  
SELECT CCODE, MAX(STDATE) AS "MOST RECENTLY STARTED"  
FROM BATCHES  
GROUP BY CCODE;  
  
# Ques no Grp-05  
SELECT SUM(AMOUNT) AS PAYMENT  
FROM PAYMENTS  
GROUP BY PAYDATE;  
  
# Ques no Grp-06  
SELECT CCODE, COUNT(BATCHCODE) AS "BATCH COUNT"  
FROM BATCHES  
GROUP BY CCODE;  
  
# Ques no Grp-07  
SELECT MONTHNAME(PAYDATE) AS MONTH, YEAR(PAYDATE), SUM(AMOUNT) AS "AMOUNT COLLECTED"  
FROM PAYMENTS  
GROUP BY MONTHNAME(PAYDATE), YEAR(PAYDATE);  
  
# Ques no Grp-08  
SELECT BATCHCODE  
FROM STUDENTS  
GROUP BY BATCHCODE  
HAVING COUNT(BATCHCODE) > 10;  
  
# Ques no Grp-09  
SELECT BATCHCODE, MIN(ADMNO), MAX(ADMNO)  
FROM STUDENTS  
GROUP BY BATCHCODE;  
  
# Ques no Grp-10  
SELECT CCODE  
FROM BATCHES  
GROUP BY CCODE, MONTH(STDATE), YEAR(STDATE)  
HAVING COUNT(BATCHCODE) > 1;  
  
# Ques no Grp-11  
SELECT CCODE, YEAR(STDATE), COUNT(BATCHCODE)  
FROM BATCHES  
GROUP BY CCODE, YEAR(STDATE);  
  
# Ques no Grp-12  
SELECT COUNT(\*)  
FROM BATCHES  
WHERE ENDDATE IS NULL  
 OR ENDDATE > CURRENT\_DATE;  
  
# Ques no Grp-13  
SELECT SUBSTRING(EMAIL, LOCATE('@', EMAIL) + 1, LENGTH(EMAIL) - 1) AS SERVER,  
 COUNT(ADMNO)  
FROM STUDENTS  
GROUP BY SERVER;  
  
# Ques no Grp-14  
SELECT COUNT(ADMNO)  
FROM STUDENTS  
GROUP BY BATCHCODE  
HAVING COUNT(BATCHCODE) > 10  
ORDER BY COUNT(BATCHCODE);  
  
# Ques no Grp-15  
SELECT MONTH(CURRENT\_DATE), SUM(AMOUNT)  
FROM PAYMENTS  
WHERE MONTH(PAYDATE) = MONTH(CURRENT\_DATE)  
GROUP BY MONTH(PAYDATE);  
# Again do we need a group by?  
  
# Ques no Grp-16  
SELECT YEAR(PAYDATE), SUM(AMOUNT) AS 'TOTAL PAYMENTS'  
FROM PAYMENTS  
GROUP BY YEAR(PAYDATE);  
  
# Ques no Grp-18  
SELECT PAYDATE  
FROM PAYMENTS  
GROUP BY PAYDATE  
HAVING SUM(AMOUNT) > 5000;  
  
# Ques no Grp-19  
SELECT BATCHCODE  
FROM STUDENTS  
WHERE TIMESTAMPDIFF(DAY, 10, CURRENT\_DATE) <= DJ  
 AND DJ < CURRENT\_DATE  
GROUP BY BATCHCODE  
HAVING COUNT(BATCHCODE) > 5;  
  
  
# Joining  
  
# Ques no Join-01  
SELECT BATCHCODE, CCODE, STDATE  
FROM BATCHES;  
  
# Ques no Join-02  
SELECT RCPTNO, FULLNAME, AMOUNT, PAYDATE  
FROM PAYMENTS P  
 INNER JOIN STUDENTS S USING (ADMNO)  
ORDER BY PAYDATE;  
  
# Ques no Join-03  
SELECT CCODE, BATCHCODE, FULLNAME  
FROM BATCHES  
 INNER JOIN STUDENTS USING (BATCHCODE);  
  
# Ques no Join-04  
SELECT CCODE, COUNT(CCODE)  
FROM BATCHES  
 INNER JOIN STUDENTS S USING (BATCHCODE)  
GROUP BY CCODE;  
  
# Ques no Join-05  
SELECT ADMNO, SUM(AMOUNT)  
FROM STUDENTS  
 INNER JOIN PAYMENTS USING (ADMNO)  
WHERE BATCHCODE = 'ora130508'  
GROUP BY ADMNO;  
  
# Ques no Join-07  
SELECT B.\*  
FROM COURSES  
 INNER JOIN BATCHES B USING (CCODE)  
WHERE COURSEFEE > 3000;  
  
# Ques no Join-08  
SELECT RCPTNO, FULLNAME, BATCHCODE, AMOUNT, PAYDATE  
FROM STUDENTS  
 INNER JOIN PAYMENTS USING (ADMNO)  
WHERE DATEDIFF(CURRENT\_DATE, PAYDATE) < 10;  
  
# Ques no Join-09  
SELECT CNAME, BATCHCODE, STDATE, ENDDATE  
FROM BATCHES  
 INNER JOIN COURSES USING (CCODE)  
WHERE ENDDATE IS NOT NULL  
 OR ENDDATE < CURRENT\_DATE;  
  
# Ques no Join-10  
SELECT FULLNAME, DJ, AMOUNT  
FROM STUDENTS  
 INNER JOIN PAYMENTS USING (ADMNO)  
WHERE DJ = PAYDATE;  
  
# Ques no Join-11  
SELECT CNAME, BATCHCODE  
FROM BATCHES  
 RIGHT JOIN COURSES USING (CCODE);  
  
# Ques no Join-12  
SELECT S.\*  
FROM STUDENTS S  
 LEFT JOIN PAYMENTS USING (ADMNO)  
WHERE AMOUNT IS NULL;  
  
# Ques no Join-13  
SELECT B.\*  
FROM BATCHES A  
 INNER JOIN BATCHES B on A.STDATE < B.STDATE  
WHERE A.BATCHCODE = 'ora130508';  
  
# Ques no Join-14  
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;  
  
# Ques no Join-15  
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  
  
# Ques no sq-01  
SELECT \*  
FROM PAYMENTS  
WHERE ADMNO = (SELECT ADMNO FROM STUDENTS WHERE FULLNAME = 'JAMES GOODWILL');  
  
# Ques no sq-02  
SELECT \*  
FROM PAYMENTS  
WHERE ADMNO IN (SELECT ADMNO FROM STUDENTS WHERE BATCHCODE = 'ORA270608');  
  
# Ques no sq-03  
SELECT BATCHCODE  
FROM BATCHES  
WHERE CCODE IN (SELECT CCODE FROM COURSES WHERE DURATION > 40);  
  
# Ques no sq-04  
SELECT FULLNAME  
FROM STUDENTS  
WHERE ADMNO IN (SELECT DISTINCT ADMNO FROM PAYMENTS WHERE MONTH(PAYDATE) = MONTH(CURRENT\_DATE));  
  
# Ques no sq-05  
SELECT FULLNAME  
FROM STUDENTS  
WHERE BATCHCODE IN (SELECT BATCHCODE FROM BATCHES WHERE CCODE LIKE '%ORACLE%');  
  
# Ques no sq-06  
SELECT \*  
FROM STUDENTS  
WHERE BATCHCODE IN (SELECT BATCHCODE FROM BATCHES WHERE ENDDATE IS NULL OR CURRENT\_DATE < ENDDATE);  
  
# Ques no sq-07  
SELECT FULLNAME, EMAIL  
FROM STUDENTS  
WHERE BATCHCODE IN (SELECT BATCHCODE FROM BATCHES WHERE MONTH(ENDDATE) = MONTH(TIMESTAMPDIFF(MONTH, 6, CURRENT\_DATE)));  
  
# Ques no sq-08  
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%');  
  
# Ques no sq-09  
SELECT BATCHCODE, COUNT(BATCHCODE) AS COUNT  
FROM STUDENTS  
GROUP BY BATCHCODE  
HAVING COUNT(BATCHCODE) < 10;  
  
# Ques no sq-10  
SELECT RCPTNO, FULLNAME, BATCHCODE, AMOUNT, PAYDATE  
FROM STUDENTS  
 INNER JOIN PAYMENTS USING (ADMNO)  
WHERE BATCHCODE IN (  
 SELECT BATCHCODE  
 FROM BATCHES  
 WHERE CCODE LIKE '%ORACLE%');  
  
# Ques no sq-11  
SELECT DISTINCT CCODE  
FROM BATCHES  
WHERE BATCHCODE IN (SELECT BATCHCODE FROM STUDENTS GROUP BY BATCHCODE HAVING COUNT(\*) > 10);  
  
# Ques no sq-12  
SELECT BATCHCODE, SUM(AMOUNT) AS COLLECTED\_AMOUNT  
FROM STUDENTS  
 INNER JOIN PAYMENTS USING (ADMNO)  
GROUP BY BATCHCODE  
HAVING SUM(AMOUNT) > 20000;  
# Cant use sub queries to add columns unless we use joins  
  
# Ques no sq-13  
SELECT FULLNAME  
FROM STUDENTS  
WHERE ADMNO NOT IN (SELECT ADMNO FROM STUDENTS WHERE DJ IN (SELECT STDATE FROM BATCHES));  
  
# Ques no sq-14  
SELECT BATCHCODE  
FROM BATCHES  
WHERE BATCHCODE IN (  
 SELECT DISTINCT BATCHCODE  
 FROM STUDENTS  
 GROUP BY BATCHCODE  
 ORDER BY COUNT(ADMNO) DESC)  
LIMIT 3;  
  
# Ques no sq-15  
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;