**Report1.**

CREATE VIEW V\_REPORT1

AS

SELECT U.FIRSTNAME,U.LASTNAME,U.EMAIL,VT.VERSION\_DESC,VT.VERSION\_START\_DATE,VT.VERSION\_END\_DATE

,P.PLAN\_TYPE,PCODE.PROMOCODEID\_PK AS PROMOCODE ,PC.CARD\_NUMBER,UP.TOTAL\_PAYMENT

FROM T\_USER U

INNER JOIN T\_BRIDGE B ON U.USERID\_PK = B.USERID\_FK

INNER JOIN T\_PLAN P ON B.PLANID\_FK = P.PLANID\_PK

LEFT JOIN T\_USER\_PAYMENT UP ON U.USERID\_PK = UP.USERID\_FK AND P.PLANID\_PK = UP.PLANID\_FK

LEFT JOIN T\_VERSION\_TYPE VT ON UP.VERSION\_TYPE\_ID\_FK = VT.VERSION\_TYPE\_ID\_PK

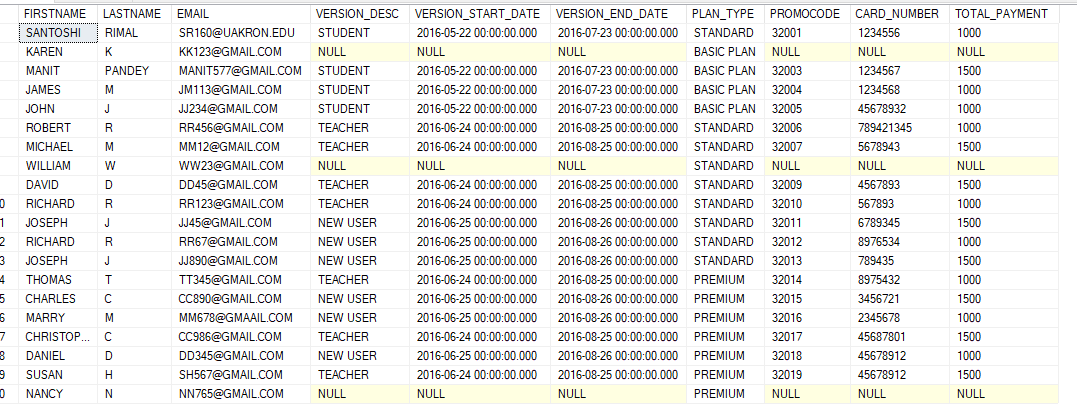
LEFT JOIN T\_PAYMENTCARD PC ON UP.PAYMENTID\_PK = PC.PAYMENTID\_PK

LEFT JOIN T\_PROMOCODE PCODE ON UP.PROMOCODEID\_FK = PCODE.PROMOCODEID\_PK

**Note:**

The View would be generated and the user should be able to run the select SQL with multiple where condition (filters).

Total\_PAYMENT is added from the previous assignment. There was a referential integrity issue between PlanID & UserID combination in T\_PLAN table and T\_USER\_PAYMNET table and the record was corrected in one of the T\_USER\_PAYMNET table to make it aligned. There are few null records in Version\_Desc and Version\_Start\_Date due to user not having any payment transaction which is expected.



**REPORT2.**

CREATE VIEW V\_REPORT2

AS

SELECT U.USERNAME,U.AGE, VT.VERSION\_DESC AS VERSION\_TYPE

,CQF.CATEGORY\_DESC,CQF.SUB\_CATEGORY\_DESC, B.FLASHCARDNUM\_FK, Q.QUIZLETNUM\_PK,G.GAME\_TYPE,G.START\_TIME

,G.END\_TIME

FROM T\_USER U

INNER JOIN T\_BRIDGE B ON U.USERID\_PK = B.USERID\_FK

INNER JOIN T\_CATEGORY\_QUIZLET\_FLASHCARD CQF ON B.CATEGORYID\_FK = CQF.CATEGORYID\_PK AND B.SUB\_CATEGORY\_ID\_FK = CQF.SUB\_CATEGORY\_ID\_PK

INNER JOIN T\_FLASHCARD F ON B.FLASHCARDNUM\_FK = F.FLASHCARDNUM\_PK AND CQF.CATEGORYID\_PK = F.CATEGORYID\_FK AND CQF.SUB\_CATEGORY\_ID\_PK = F.SUB\_CATEGORY\_ID\_FK

INNER JOIN T\_QUIZLET Q ON B.QUIZLETNUM\_FK = Q.QUIZLETNUM\_PK AND CQF.CATEGORYID\_PK = Q.CATEGORYID\_FK AND CQF.SUB\_CATEGORY\_ID\_PK = Q.SUB\_CATEGORY\_ID\_FK

INNER JOIN T\_GAME G ON B.GAME\_ID\_FK = G.GAME\_ID\_PK

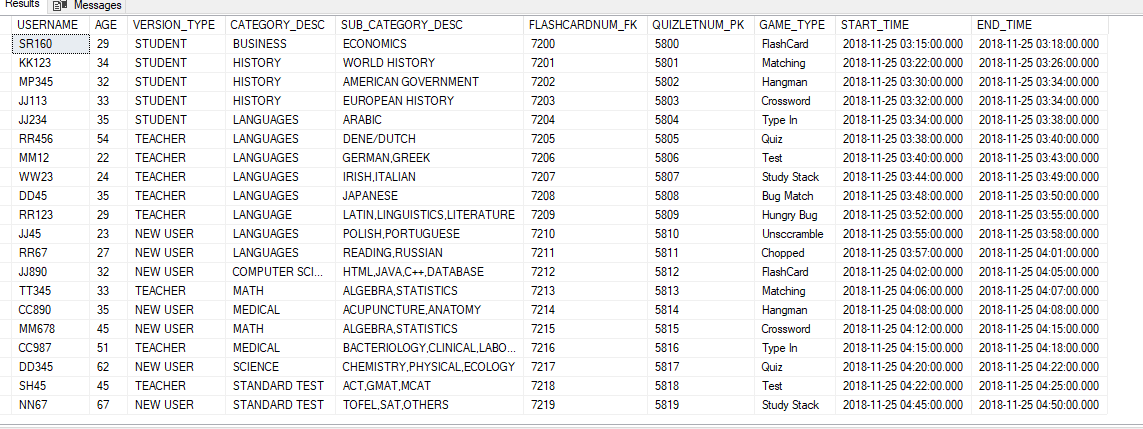
LEFT JOIN T\_USER\_PAYMENT UP ON U.USERID\_PK=UP.USERID\_FK

LEFT JOIN T\_VERSION\_TYPE VT ON UP.VERSION\_TYPE\_ID\_FK=VT.VERSION\_TYPE\_ID\_PK

**Note:**

After creating this view user should be able to run the report to see what kinds of customer are using what kinds of products based upon different filters like version\_Desc, Game\_Type.

There was also a referential integrity issues between different combinations between different tables. Records had to be corrected in Bridge table to make it same Categoryid and Sub\_Category\_id to make it similar with T\_caegory\_quizlet\_flashcard table and again the T\_QUIZLET table has been edited to make the QUIZLETNUM, Categoryid and Sub\_category\_id similar in bridge table.



**Report 3.**

CREATE VIEW V\_REPORT3

AS

SELECT PT.PRO\_ID,U.USERID\_PK,U.USERNAME,U.FIRSTNAME,U.LASTNAME,Q.QUIZLETNUM\_PK

,W.TERM,G.GAME\_TYPE,PT.START\_TIME,PT.END\_TIME,G.RESULTS,

CASE WHEN CAST(G.RESULTS AS INT) BETWEEN 70 AND 79 THEN 'C' WHEN CAST(G.RESULTS AS INT) BETWEEN 80 AND 89 THEN 'B' WHEN CAST(G.RESULTS AS INT) BETWEEN 90 AND 100 THEN 'A' ELSE 'FAIL' END AS SCORE

FROM T\_PRO\_TEACHER\_TABLE PT

INNER JOIN T\_USER U ON U.USERID\_PK=PT.USERID\_FK

INNER JOIN T\_BRIDGE B ON U.USERID\_PK=B.USERID\_FK

LEFT JOIN T\_QUIZLET Q ON PT.QUIZLETNUM\_FK = Q.QUIZLETNUM\_PK

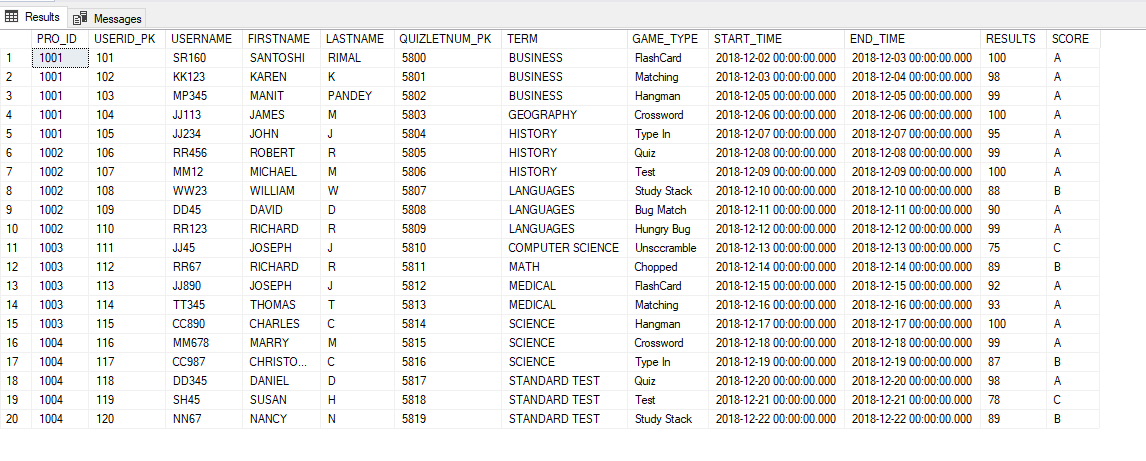
LEFT JOIN T\_GAME G ON B.GAME\_ID\_FK=G.GAME\_ID\_PK

LEFT JOIN T\_WORD W ON G.WORD\_ID\_FK = W.WORD\_ID\_PK

**Note:**

After creating this report view the user (teacher) should be able to see how many quiz and scores their students achieved. The cast and case statement is used to convert the student result into score (i.e. A,B,C)

There was a mistake while defining the field PRO\_ID in table T\_PRO\_TEACHER, it would have multiple userids (students) associated under one PRO\_ID and no longer would be a primary key, the whole table had to be dropped and create a new table without primary key and all new data were inserted making sure no referential integrity would be violated



**Report 4.**

CREATE VIEW V\_REPORT4

AS

SELECT A.ADVERTISEMENT\_ID\_PK,A.ADVERTISEMENT\_STATE,A.STARTTIME,

A.ENDTIME,A.NUM\_VIEWED,AP.VENDOR\_NAME,AP.TOTAL\_PAYMENT\_AMOUNT

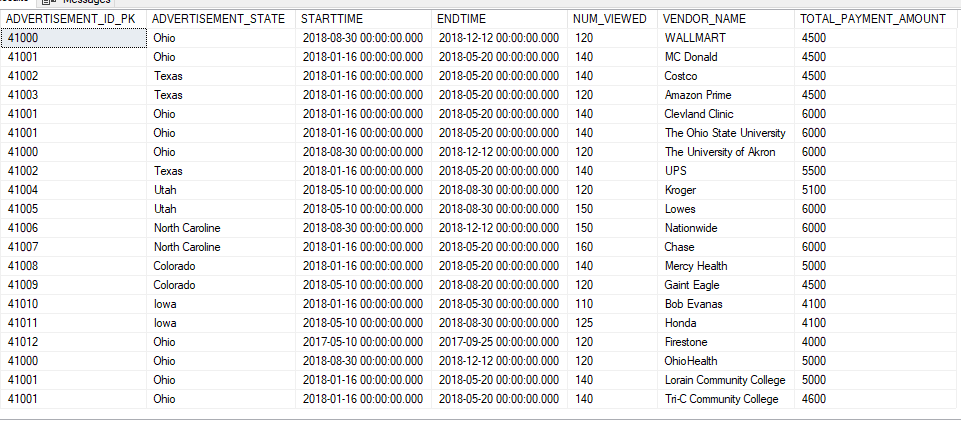
FROM T\_ADVERTISEMENT A INNER JOIN

T\_ADVERTISEMENT\_PAYMENT AP ON A.ADVERTISEMENT\_ID\_PK =AP.ADVERTISEMENT\_ID\_FK

**Note:**

This was a simple report to focus on the advertisement, after creating the view report, user should be able to filter on different state.

There were no other issues detected while creating this report.



**Report 5.**

CREATE VIEW V\_REPORT5

AS

SELECT C.CREATOR\_ID\_PK,F.FLASHCARDNUM\_PK,W.WORD\_ID\_PK,G.GAME\_ID\_PK,G.START\_TIME,G.END\_TIME,U.USERID\_PK

FROM T\_CREATOR C

INNER JOIN T\_BRIDGE B ON C.CREATOR\_ID\_PK = B.CREATOR\_ID\_FK

INNER JOIN T\_USER U ON B.USERID\_FK = U.USERID\_PK

LEFT JOIN T\_FLASHCARD F ON B.FLASHCARDNUM\_FK = F.FLASHCARDNUM\_PK

LEFT JOIN T\_WORD W ON C.CREATOR\_ID\_PK = W.CREATOR\_ID\_FK

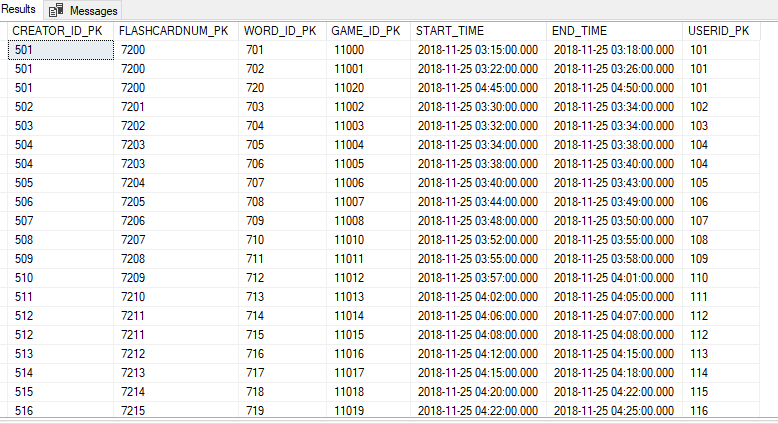
LEFT JOIN T\_GAME G ON W.WORD\_ID\_PK = G.WORD\_ID\_FK

**Note:**

After creating this view the user should be able to filter report based upon their flashacards id and see how it is doing.

The FLASHCARDNUM\_PK table was missing on the previous assignment in part of required tables and fields, and had to be added again everything was looking good.

Few of the word table related records are coming Null in the view but that was expected; some creator has just created the account and hasn’t done anything further.



**Report 6.**

CREATE VIEW V\_REPORT6

AS

SELECT G.GAME\_ID\_PK,B.USERID\_FK,G.START\_TIME,G.END\_TIME

FROM T\_GAME G

INNER JOIN T\_BRIDGE B ON G.GAME\_ID\_PK = B.GAME\_ID\_FK

**Note:**

After creating this report the user can filter their game to see how it is doing. There was no issue found.

