Andrew ID : ruixinh Name: Ruixin Huang

SQL> START C:\Users\Ray\Desktop\CMULectureSpring\database\HW\SQLHW2\HW2.TXT

SQL> SET LINESIZE 230;

SQL> SET PAGESIZE 200;

SQL>

SQL> COLUMN CATALOG\_NUM FORMAT A10;

SQL> COLUMN TITLE FORMAT A50;

SQL>

SQL> --Part I--

SQL> --Q1:--

SQL> UPDATE INVENTORY

2 SET NEW = 'N'

3 WHERE CATALOG\_NUM IN

4 (SELECT CATALOG\_NUM

5 FROM FILM JOIN INVENTORY USING(CATALOG\_NUM)

6 WHERE YEAR<2015 AND NEW = 'Y');

70 rows updated.

SQL>

SQL> --Q2:--

SQL> ALTER TABLE INVENTORY

2 DROP CONSTRAINT INVENTORY\_CATALOG\_FK;

Table altered.

SQL>

SQL> ALTER TABLE FILM

2 DROP CONSTRAINT FILM\_PK;

Table altered.

SQL>

SQL> UPDATE INVENTORY

2 SET CATALOG\_NUM = (

3 SELECT SUBSTR(Catalog\_Num,1,LENGTH(CATALOG\_NUM)-1)||SUBSTR(FORMAT,1,1)

4 FROM FILM

5 WHERE FILM.CATALOG\_NUM=INVENTORY.CATALOG\_NUM);

102 rows updated.

SQL>

SQL> UPDATE FILM

2 SET Catalog\_Num = CONCAT(SUBSTR(Catalog\_Num,1,LENGTH(CATALOG\_NUM)-1),SUBSTR(FORMAT,1,1));

45 rows updated.

SQL>

SQL> ALTER TABLE FILM

2 ADD CONSTRAINT FILM\_PK

3 PRIMARY KEY(CATALOG\_NUM);

Table altered.

SQL>

SQL> ALTER TABLE INVENTORY

2 ADD CONSTRAINT INVENTORY\_CATALOG\_FK

3 FOREIGN KEY (CATALOG\_NUM)

4 REFERENCES FILM (CATALOG\_NUM);

Table altered.

SQL>

SQL> --Q3:--

SQL> ALTER TABLE FILM

2 ADD

3 CONSTRAINT FILM\_VALUE\_VALID

4 CHECK(SUBSTR(Catalog\_Num,LENGTH(CATALOG\_NUM),1)=SUBSTR(FORMAT,1,1));

Table altered.

SQL>

SQL> --Q4:--

SQL> ALTER TABLE RENTED\_ITEM

2 ADD

3 CONSTRAINT LATE\_FEE\_REQ

4 CHECK((DUE\_DATE>=RETURN\_DATE AND LATE\_FEE IS NULL) OR (DUE\_DATE<RETURN\_DATE AND LATE\_FEE IS NOT NULL));

Table altered.

SQL>

SQL> --DESCRIPTION: We need to ensure the Late\_fee exists only happens when return date later than due date.--

SQL>

SQL> ALTER TABLE MANAGER

2 ADD

3 CONSTRAINT TRAIN\_DATE\_VALID

4 CHECK (LAST\_TRAINING >= BEGIN\_DATE);

Table altered.

SQL>

SQL> --DESCRIPTION: We need to ensure member's last traing date happens later than his/her first training.--

SQL>

SQL> --Part II--

SQL>

SQL> --Q1:--

SQL> SELECT TITLE, YEAR,COUNT(DISTINCT FILM\_ID) AS "Number of Copies",COUNT(RENTAL\_NUM) AS "Rented Times"

2 FROM FILM JOIN INVENTORY USING (CATALOG\_NUM)

3 LEFT OUTER JOIN RENTED\_ITEM USING (FILM\_ID)

4 WHERE GENRE LIKE '%Sci-Fi%'

5 GROUP BY TITLE,YEAR

6 ORDER BY "Rented Times" DESC;

TITLE YEAR Number of Copies Rented Times

-------------------------------------------------- ---------- ---------------- ------------

Alphas: Season 2 2013 9 5

Avatar 2010 2 3

Alphas: Season 1 2012 4 2

Guardians of the Galaxy 2014 5 2

Star Trek 2010 9 2

Star Trek: Into Darkness 2013 2 1

Star Wars: The Force Awakens 2016 1 0

Rogue One: A Star Wars Story 2017 1 0

8 rows selected.

SQL>

SQL> --Q2:--

SQL> SELECT EMP\_ID||' '||FIRST\_NAME||' '||LAST\_NAME AS "SALES\_REP\_INFO",

2 COUNT(DISTINCT RENTAL\_NUM) AS "Number of Rental",

3 COUNT(FILM\_ID) AS "Number of Films"

4 FROM EMPLOYEE JOIN SALES\_REP USING (EMP\_ID)

5 LEFT OUTER JOIN RENTAL USING (EMP\_ID)

6 LEFT OUTER JOIN RENTED\_ITEM USING (RENTAL\_NUM)

7 GROUP BY EMP\_ID,FIRST\_NAME,LAST\_NAME

8 ORDER BY "Number of Films" DESC;

SALES\_REP\_INFO Number of Rental Number of Films

----------------------------------------------- ---------------- ---------------

105 Gloria Tremblay 5 13

106 Emma Wesley 5 12

101 Jim Smith 4 8

104 Victoria Lee 3 7

103 Lily Evans 3 7

102 Steve Baker 2 6

107 Luis Suarez 0 0

7 rows selected.

SQL>

SQL> --Q3:--

SQL> COLUMN "MEMBER\_INFO" FORMAT A20;

SQL> COLUMN RENTAL\_NUM FORMAT A10;

SQL> COLUMN CATALOG\_NUM FORMAT A11;

SQL> COLUMN TITLE FORMAT A50;

SQL> COLUMN "INDICATOR" FORMAT A10;

SQL> COLUMN "RENT\_DATE" FORMAT 999999999;

SQL>

SQL> SELECT MEMBER\_ID||' '|| FIRST\_NAME || ' ' || LAST\_NAME AS "MEMBER\_INFO",

2 RENTAL\_NUM,RENT\_DATE,CATALOG\_NUM,TITLE,

3 (CASE WHEN RETURN\_DATE IS NULL AND LATE\_FEE IS NOT NULL

4 THEN 'OVERDUE'

5 WHEN RETURN\_DATE IS NOT NULL AND LATE\_FEE IS NOT NULL

6 THEN 'LATE'

7 WHEN RETURN\_DATE IS NULL AND LATE\_FEE IS NULL

8 THEN 'NOT RETURN'

9 ELSE 'ON TIME'

10 END) AS "INDICATOR",

11 (CASE WHEN RETURN\_DATE IS NULL THEN SYSDATE-RENT\_DATE

12 WHEN RETURN\_DATE IS NOT NULL THEN RETURN\_DATE-RENT\_DATE

13 END) AS "RENT\_DATE"

14 FROM FILM JOIN INVENTORY USING (CATALOG\_NUM)

15 JOIN RENTED\_ITEM USING (FILM\_ID)

16 JOIN RENTAL USING (RENTAL\_NUM)

17 JOIN MEMBER USING (MEMBER\_ID)

18 JOIN STORE ON STORE.STORE\_NUM = MEMBER.STORE\_NUM

19 WHERE RENT\_DATE>=(SYSDATE-30) AND INVENTORY.STORE\_NUM = 10;

MEMBER\_INFO RENTAL\_NUM RENT\_DATE CATALOG\_NUM TITLE INDICATOR RENT\_DATE

-------------------- ---------- --------- ----------- -------------------------------------------------- ---------- ----------

004 Muriel Green 14 13-MAR-19 64W The Avengers ON TIME 7

004 Muriel Green 14 13-MAR-19 65W Wrath of the Titans ON TIME 8

001 David Lee 18 15-MAR-19 68W Alphas: Season 2 ON TIME 6

001 David Lee 18 15-MAR-19 71W The Croods ON TIME 5

SQL> --Q4:----

SQL> COLUMN "DAYS OUT" FORMAT A10;

SQL>

SQL> SELECT CATALOG\_NUM,MEMBER\_ID||' '|| FIRST\_NAME || ' ' || LAST\_NAME AS "MEMBER\_INFO",

2 RENT\_DATE,FILM\_ID,NUMBER\_DISCS,DUE\_DATE,RETURN\_DATE,

3 (CASE WHEN RETURN\_DATE IS NULL THEN 'NA'

4 WHEN RETURN\_DATE IS NOT NULL THEN TO\_CHAR(RETURN\_DATE-RENT\_DATE)

5 END) AS "DAYS OUT",RENTAL\_COST

6 FROM FILM JOIN INVENTORY USING (CATALOG\_NUM)

7 JOIN RENTED\_ITEM USING (FILM\_ID)

8 JOIN RENTAL USING (RENTAL\_NUM)

9 JOIN MEMBER USING (MEMBER\_ID)

10 JOIN

11 (SELECT RENTAL\_NUM,SUM(PRICE+NVL(LATE\_FEE,0)) RENTAL\_COST

12 FROM FILM JOIN INVENTORY USING (CATALOG\_NUM)

13 JOIN RENTED\_ITEM USING (FILM\_ID)

14 JOIN RENTAL USING (RENTAL\_NUM)

15 JOIN MEMBER USING (MEMBER\_ID)

16 GROUP BY RENTAL\_NUM

17 ) USING (RENTAL\_NUM)

18 WHERE RENT\_DATE BETWEEN '15-MAR-2019' AND LAST\_DAY(TO\_DATE('MAR-2019','MON-YYYY'));

CATALOG\_NUM MEMBER\_INFO RENT\_DATE FILM\_ NUMBER\_DISCS DUE\_DATE RETURN\_DA DAYS OUT RENTAL\_COST

----------- -------------------- --------- ----- ------------ --------- --------- ---------- -----------

66M 009 Robin Singh 15-MAR-19 9967 6 29-MAR-19 19-MAR-19 4 12

68W 005 Cathy Brooks 18-MAR-19 9983 3 01-APR-19 20-MAR-19 2 9

50W 008 Tiffany Foster 20-MAR-19 9938 1 03-APR-19 NA 9

49W 009 Robin Singh 20-MAR-19 9929 1 03-APR-19 NA 9

46W 009 Robin Singh 15-MAR-19 9917 1 29-MAR-19 18-MAR-19 3 12

31W 010 Serena Sharma 15-MAR-19 4402 3 29-MAR-19 21-MAR-19 6 3

68W 001 David Lee 15-MAR-19 9977 3 29-MAR-19 21-MAR-19 6 6

61M 009 Robin Singh 20-MAR-19 9946 1 03-APR-19 NA 9

68W 002 Jose Sanchez 20-MAR-19 9985 3 03-APR-19 NA 6

67W 002 Jose Sanchez 20-MAR-19 9976 1 03-APR-19 NA 6

68W 008 Tiffany Foster 20-MAR-19 9980 3 03-APR-19 NA 9

68W 003 Justin Parker 15-MAR-19 9981 3 29-MAR-19 NA 9

71W 001 David Lee 15-MAR-19 9986 1 29-MAR-19 20-MAR-19 5 6

67W 009 Robin Singh 15-MAR-19 9974 1 29-MAR-19 18-MAR-19 3 12

73W 005 Cathy Brooks 18-MAR-19 9992 1 01-APR-19 NA 9

67W 009 Robin Singh 20-MAR-19 9973 1 03-APR-19 NA 9

60W 003 Justin Parker 15-MAR-19 9945 1 29-MAR-19 NA 9

40W 003 Justin Parker 15-MAR-19 9905 1 29-MAR-19 21-MAR-19 6 9

74W 005 Cathy Brooks 18-MAR-19 9994 1 01-APR-19 NA 9

72W 008 Tiffany Foster 20-MAR-19 9990 1 03-APR-19 20-MAR-19 0 9

41M 009 Robin Singh 15-MAR-19 1103 1 29-MAR-19 19-MAR-19 4 12

21 rows selected.

SQL>

SQL> --Q5:--

SQL> SELECT STORE\_NUM,COUNT(DISTINCT FILM\_ID) AS "Number of Rented",

2 COUNT(DISTINCT EMPLOYEE.EMP\_ID) AS "Number of Employee",

3 COUNT(DISTINCT FILM\_ID)/COUNT(DISTINCT SALES\_REP.EMP\_ID) AS "Avg Film Rented PR"

4 FROM RENTED\_ITEM JOIN INVENTORY USING (FILM\_ID)

5 LEFT OUTER JOIN RENTAL USING(RENTAL\_NUM)

6 LEFT OUTER JOIN STORE USING (STORE\_NUM)

7 LEFT OUTER JOIN EMPLOYEE USING (STORE\_NUM)

8 LEFT OUTER JOIN SALES\_REP ON EMPLOYEE.EMP\_ID = SALES\_REP.EMP\_ID

9 WHERE EXTRACT(YEAR FROM RENT\_DATE)=EXTRACT(YEAR FROM SYSDATE)

10 GROUP BY STORE\_NUM

11 UNION ALL

12 SELECT 'Grand Total',COUNT(DISTINCT FILM\_ID),COUNT(DISTINCT EMPLOYEE.EMP\_ID),COUNT(DISTINCT FILM\_ID)/COUNT(DISTINCT SALES\_REP.EMP\_ID)

13 FROM RENTED\_ITEM JOIN INVENTORY USING (FILM\_ID)

14 LEFT OUTER JOIN RENTAL USING(RENTAL\_NUM)

15 LEFT OUTER JOIN STORE USING (STORE\_NUM)

16 LEFT OUTER JOIN EMPLOYEE USING (STORE\_NUM)

17 LEFT OUTER JOIN SALES\_REP ON EMPLOYEE.EMP\_ID = SALES\_REP.EMP\_ID

18 WHERE EXTRACT(YEAR FROM RENT\_DATE)=EXTRACT(YEAR FROM SYSDATE);

STORE\_NUM Number of Rented Number of Employee Avg Film Rented PR

----------- ---------------- ------------------ ------------------

10 18 4 6

20 16 3 8

30 12 3 6

Grand Total 46 10 6.57142857

SQL>

SQL>

SQL>

SQL>

SQL>

SQL>

SQL>

SQL>

SQL>

SQL>

SQL>

SQL>

SQL>

SQL> --Q6:--

SQL> SELECT STORE\_NUM||' '||STORE\_NAME AS "STORE\_INFO",COUNT(DISTINCT FILM\_ID) AS "RENTED\_NUM",SUBSTR((COUNT(DISTINCT FILM\_ID)/STORE\_INVENTORY)\*100,1,4)||'%' AS "PERCENTAGE\_INVENTORY"

2 FROM INVENTORY LEFT OUTER JOIN RENTED\_ITEM USING (FILM\_ID)

3 LEFT OUTER JOIN RENTAL USING (RENTAL\_NUM)

4 LEFT OUTER JOIN STORE USING (STORE\_NUM)

5 JOIN

6 (SELECT STORE\_NUM,COUNT(DISTINCT FILM\_ID) STORE\_INVENTORY

7 FROM INVENTORY LEFT OUTER JOIN RENTED\_ITEM USING (FILM\_ID)

8 LEFT OUTER JOIN RENTAL USING (RENTAL\_NUM)

9 LEFT OUTER JOIN STORE USING (STORE\_NUM)

10 GROUP BY STORE\_NUM)

11 USING (STORE\_NUM)

12 WHERE RETURN\_DATE IS NULL AND RENT\_DATE IS NOT NULL

13 GROUP BY STORE\_NUM,STORE\_NAME,STORE\_INVENTORY;

STORE\_INFO RENTED\_NUM PERCENTAGE\_INVENT

------------------------------------ ---------- -----------------

10 Oakland's DVD 1 2.32%

30 DVD Hub 5 20.8%

20 SQ Hill's DVD 8 22.8%

SQL>

SQL>

SQL>

SQL> --Q7:--

SQL> SELECT GENRE,

2 COUNT(DISTINCT CATALOG\_NUM) AS "Number Of Films",

3 COUNT(RENTAL\_NUM) AS "Rented Times"

4 FROM FILM LEFT OUTER JOIN INVENTORY USING (CATALOG\_NUM)

5 LEFT OUTER JOIN RENTED\_ITEM USING (FILM\_ID)

6 GROUP BY GENRE

7 UNION ALL

8 SELECT 'Grand Total',

9 COUNT(DISTINCT CATALOG\_NUM) AS "Number Of Films",

10 COUNT(RENTAL\_NUM) AS "Rented Times"

11 FROM FILM LEFT OUTER JOIN INVENTORY USING (CATALOG\_NUM)

12 LEFT OUTER JOIN RENTED\_ITEM USING (FILM\_ID);

GENRE Number Of Films Rented Times

------------------------- --------------- ------------

Comedy 2 3

Sci-Fi 8 8

TV Drama 5 1

Action and Adventure 9 15

Children and Family 5 5

Thriller 1 1

Classic, Drama 1 0

Classic, War 1 1

Classic 4 3

Fantasy 4 7

TV Sci-Fi 2 7

Documentary 1 1

War 1 1

Classic, Romance 1 0

Grand Total 45 53

15 rows selected.

SQL> SPOOL OUT