

**CALIFORNIA STATE UNIVERSITY, SACRAMENTO**  
College of Business Administration

MIS 150 - Database Management Systems for Business

**Relational Algebra and SQL Homework Assignment**

Points: 45

Due: Monday, October 10

The objectives of this assignment are to familiarize you with SQL, relate the relational algebra presented in class and your textbook to basic SQL queries, and to provide you with experience in Oracle9i. It involves creating a user and two (database) tables, loading the tables with data from *control files*, importing a third table, and performing 14 queries on them. Follow the Camtasia videos for logging into SQL Plus, creating and importing the tables, and building SQL queries. You will need to capture your session in a spool file to receive credit (see class notes).

**Suggestion.** *Plan ahead!* Read chapters 4, 5 and 6 before attempting this assignment to avoid wasting many hours.

**Creating a User**

Create user MIS150 with a password of MIS150 (or anything else) in SQL Plus. Grant your user a DBA role so you may create and import tables. The syntax for creating a user is as follows:

```
CREATE USER username IDENTIFIED BY password;
```

where you supply the *username* and *password*.

**Note.** All SQL commands (statements) must be terminated by a semicolon (;).

To grant your user a DBA role, enter:

```
GRANT dba TO username;
```

You may use this user name and password for subsequent logons. To test you username and password, disconnect your current user

```
DISCONNECT
```

and reconnect with your new one

```
CONNECT username
```



To delete a user, issue the following command:

```
DROP USER username;
```

## Defining the Tables

Create two tables, VIDEO\_CATEGORIES and VIDEOS. Use SQL Plus to define the VIDEO\_CATEGORIES and VIDEOS tables with the (SQL) CREATE TABLE command. The following tables list the column names, data types, lengths and constraints. Since data will be loaded into the tables from control files (videos.ctl, category.ctl), the column names must be identical to those listed below. Otherwise, you will have problems loading data into them.

**Note.** You will need to capture both your CREATE TABLE in a spool file to receive credit.

### Video\_Categories

Column Name	Data Type	Length	Integrity Constraint
Category_code	char	2	primary key
Category_title	varchar2	40	not null

### Videos

Column Name	Data Type	Length	Integrity Constraints
Stock_number	char	6	primary key
Video_title	varchar2	45 50	not null
Retail_price	decimal		default = 0
Running_time	<del>decimal</del> small integer		default = 0
Category_code	char	2	foreign key
Rating	varchar2	5	default = N/A
Distributor_code	varchar2	3	foreign key
Year_released	date		(none)
Active_date	date		(none)



Stock_on_hand	integer		default = 0
Stock_on_order	integer		default = 0

### Importing a Table

The third table, DISTRIBUTORS, has been exported to a dump (dmp) file, H1.DMP (compressed in H1.exe). Use Oracle's import utility IMP to import this table into your database. Both the table's structure and data will be inserted to the database. Thus, the table's definition nor data are of concern.

First, create a DOS window and change to the directory (folder) with the dump file. To open a DOS window

1. Click on "Start" in the lower left corner of Windows XP.
2. Click on "Run..."
3. In the Run dialog box, enter "command" and click on "OK."

The general syntax for using Importer is:

```
imp user-name/password
```

where:      *user-name* and *password*      User name and password assign to the user (e.g., mis150). Be sure to separate the name and password with the "/".

Afterwards, view the table's definition through the DESCRIBE command in SQL Plus:

```
describe table-name
```

### Loading Data into the Tables

Once the VIDEO\_CATEGORIES and VIDEOS tables have been defined, they can be loaded with data from two control files, category.ctl and videos.ctl, respectively (both are contained in h1.exe). Minimize the SQL Plus window (do NOT exit SQL Plus) and create a DOS window. Change to the drive/directory under which you have placed these files.

The general syntax to enter at the DOS prompt is as follows:

```
sqlldr user-name/password file-name
```



where:	<i>user-name</i> and <i>password</i>	User name and password assign to the user (e.g., mis150). Be sure to separate the name and password with the “/”.
	<i>file-name</i>	The name of the control file (either categories or videos).

Do this for both files. Before exiting the DOS window (i.e., enter EXIT and press the ENTER key) and returning to SQL Plus, examine the *log* files (category.log, videos.log) to see if all records have been loaded into the tables (video\_categories should have 22 rows and videos 248). If they have not, resolve all conflicts before reloading the table. In most cases, the problem will stem from a misspelled column name or an incorrectly specified column width. To view the records that were not loaded, open the *bad* file (categories.bad, videos.bad) in a text editor.

**Note.** Before attempting this, download and view the Camtasia videos.

### Dropping a Table

In the event you need to drop your table, use the DROP TABLE (SQL) command:

```
DROP TABLE table_name;
```

### Queries

Now that all tables contain data, you can perform the following queries. The details of the queries (i.e., predicate, column names) are of your own doing. Thus, it would be uncanny for two assignments to contain identical entries. Also, be creative! Do not closely follow the class notes or Camtasia videos. Doing so may be a detriment to your grade.

**Note.** When designing your queries, be sure they convey **usable information**. Queries that are comprised of a nonsensical assembly of columns and/or rows will not awarded full points. For example, constructing a query that only lists prices or retrieves no rows is not useful information. Look at your output and ask yourself if someone else can understand it.

Please apply your *common sense!* *This isn't just an exercise, it's an investment in your future!*

**Carefully read the relational algebra!**

1. Selection. Perform a selection on the videos table. The number of rows returned by the query should not exceed 15.



$$S_{predicate}(videos)$$

Refer to the section beginning on page 4 (Comparison Operators) of the “Oracle SQL Reference” document for further discussion.

2. Projection and Selection. Perform a project on your selection in (1).

$$?_{column-list}(S_{predicate}(videos))$$

3. Projection and Selection using a logical AND and/or a logical OR. Add a logical AND and/or OR to your predicate in (2). The number of rows returned by the query should not exceed 10.

$$?_{column-list}(S_{condition1 \{AND | OR\} condition2}(videos))$$

Refer to the section beginning on page 8 (Logical Operators) of the “Oracle SQL Reference” document for further discussion.

4. Equi-join. Expand your query in (2) and perform an equi-join with the `videos_categories` or `distributors` table.

**Note.** The  $\times$  notation below means join.

$$?_{column-list}(S_{predicate}(videos)) \times_{videos.column-name=video\_categories.column-name} ?_{column-list}(video\_categories)$$

or

$$?_{column-list}(S_{predicate}(videos)) \times_{videos.column-name=distributors.column-name} ?_{column-list}(distributors)$$

5. Join and Range. Perform another join between the `videos` table and either the `distributors` or `videos_categories` table. Incorporate into your predicate a condition using the BETWEEN clause. The number of rows returned by the query should not exceed 15.

$$?_{column-list}(S_{predicate}(videos)) \times_{videos.column-name=video\_categories.column-name} ?_{column-list}(videos\_categories)$$

or

$$?_{column-list}(S_{predicate}(videos)) \times_{videos.column-name=distributors.column-name} ?_{column-list}(distributors)$$

6. Join and Pattern Matching. Replace the BETWEEN in query (5) with the LIKE clause and the UPPER/LOWER function in the predicate. The number of rows returned by the query should not exceed 15.

$$?_{column-list}(S_{predicate}(videos)) \times_{videos.column-name=video\_categories.column-name} ?_{column-list}(video\_categories)$$

or



? column-list(S predicate(*videos*)) ><\_videos.column-name=distributors.column-name ? column-list(*distributors*)

Refer to the sections beginning on pages 6 (Like Operator), 20 (LOWER) and 30 (UPPER) of the “Oracle SQL Reference” document for further discussion.

**Note.** Pattern matching is performed on character strings (data) and not numeric data. Performing a pattern match on numeric data is inappropriate (and a detriment to your grade).

7. Aggregation and GROUP BY / HAVING. Perform a query using the COUNT, SUM, AVG, MAX and MIN aggregation functions. Apply them over selected groups of entities (i.e., rows) that share a common attribute value. Also, apply a format mask to all numeric columns (refer to the Oracle online documentation). Include a comma (,) between the thousands and hundreds places (i.e., 9,999). For monetary values, include the dollar sign (\$) and decimal point. The number of rows returned by the query should exceed two (i.e., three or more) but be less than 16.

? column-list(S predicate(*videos*)) ><\_videos.column-name=video\_categories.column-name ? column-list(*video\_categories*)  
or  
? column-list(S predicate(*videos*)) ><\_videos.column-name=distributors.column-name ? column-list(*distributors*)

Refer to the sections beginning on pages 55 (Group Function) and 60 (Format Models) of the “Oracle SQL Reference” document for further discussion.

8. Calculation and SYSDATE. Perform a query involving a calculation with SYSDATE (system date) and either the date\_released or active\_date column (of videos) to determine the number of years and months between the two dates. Apply a date format mask to the date column you have selected and display its retained value; do NOT use the default mask. Attach an alias to the calculation and to\_char columns. Include other descriptive titles in your query. The number of rows returned by the query should exceed two (i.e., three or more) but be less than 16.

? column-list(S predicate(*videos*)) ><\_videos.column-name=video\_categories.column-name ? column-list(*video\_categories*)  
or  
? column-list(S predicate(*videos*)) ><\_videos.column-name=distributors.column-name ? column-list(*distributors*)

Refer to the sections beginning on pages 67 (Date Format Models), 33 (Date Functions), 17 (MOD), and 20 (TRUNC) of the “Oracle SQL Reference” document for further discussion. You may have to use the TRUNC (prevents rounding) and MOD (modulus; retains the remainder in division) functions. The syntax for the two are as follows:

TRUNC ( *arithmetic-operation* , *decimal-places* )



where: *arithmetic-operation* Composition of arithmetic operations (e.g., retail - cost / cost) or an aggregation function (e.g., avg(retail\_price)).

*decimal-places* Number of decimals to be displayed. This can be any positive integer or zero. When not included, zero is assumed.

MOD(*numerator*, *denominator*)

9. View. Create two union compatible views using projections on selections and equi-joins between the videos and either the video\_categories or distributors tables. The number of columns specified in the projections for each view may vary. However, they (projections) should include at least (i.e., a minimum of) two common columns (i.e., union compatible). When designing the selections, allow for an overlap of entities between the views. Thus, some entities will appear in both views.

$$\begin{aligned} & \text{view-name} = \\ & ? \text{column-list}(\text{S predicate}(\text{videos})) \times_{\text{videos.column-name}=\text{video\_categories.column-name}} ? \text{column-list}(\text{video\_categories}) \\ & \text{or} \\ & ? \text{column-list}(\text{S predicate}(\text{videos})) \times_{\text{videos.column-name}=\text{distributors.column-name}} ? \text{column-list}(\text{distributors}) \end{aligned}$$

Display the contents (all columns and rows) of both views with the SQL SELECT command. The number of rows produced in the view should not exceed 10. Do NOT apply any conditions to limit the number of rows or restrictions on the number of columns displayed; display everything.

**Note.** Before creating the views, study the requirements of (10) through (12).

10. Union. Perform a union on the views in (9). The number of rows produced by the union should be less than the sum of the two views (since duplicates only appear once).

$$? \text{column-list}(\text{view1}) \cup ? \text{column-list}(\text{view2})$$

Refer to the section beginning on page 10 (Set Operators) of the “Oracle SQL Reference” document for further discussion.

11. Intersection. Find the videos both views in (9) share through their intersection. There should be at least (i.e., a minimum of) 3 rows.

$$? \text{column-list}(\text{view1}) \cap ? \text{column-list}(\text{view2})$$


12. Set Difference. Find videos that are unique to one view. There should be at least (i.e., a minimum of) 3 rows.

$$?_{column-list(view1)} - ?_{column-list(view2)}$$

13. ANY/ALL with a Subquery. Perform a subquery that incorporates either the ANY or ALL. Use either the videos, video\_categories, distributors tables and/or your views in (9). Explain the purpose of your subquery and the information it presents next to your output.

```
SELECT column-list-1 FROM {table-1 | view-1}
WHERE column-name-1 {relational operator} {ANY | ALL }
  (SELECT column-list-2 FROM {table-2 | view-2}
   WHERE condition)
```

**Note.** Be sure to read pages 138-139 in your textbook before attempting this.

14. EXIST/NOT EXIST with a Subquery. Following the table/view specifications in (13), create a EXIST/NOT EXIST in your subquery. Explain the purpose of your subquery and the information it presents next to your output.

```
SELECT column-list-1 FROM {table-1 | view-1}
WHERE {EXISTS | NOT EXISTS}
  (SELECT { * | column-list-2 } FROM {table-2 | view-2}
   WHERE condition)
```

**Note.** Be sure to read pages 146-147 in your textbook before attempting this.

## Tangibles

To receive credit for this assignment, submit a printed copy of your spool file<sup>1</sup> in a 9 × 12-inch manila envelope; the electronic file is NOT required. The listing should include the CREATE TABLE for VIDEOS, and queries (1) through (14). Tab the pages with your queries, highlight the SQL command (with a color marker, preferable yellow), and write the number corresponding to the requirement next to the query. These indicate the *attempt* you want graded. As stated in the past, credit cannot be awarded for assumed work. **Unmarked work will not be graded.**

---

<sup>1</sup> Open the spool file in a word processing program, such as Microsoft Word or Notepad, and print it. Set the page orientation to landscape to prevent wrapping.





**Note.** Do not worry about errors in your spool file listing. This provides you with more evidence that the work is yours! Also, multiple sessions are acceptable.

Please also be aware of the *one assignment, one grade* rule. Any assignments that have an uncanny resemblance will be considered a violation of ethical class behavior. You may work with others, but the assignment you submit should be the product of your own work.

As stated in the syllabus, homework assignments are due at the beginning of class. Late assignments will NOT be accepted.

### **Dropping Your Tables**

If you are working on a public access computer (i.e., one in a lab), remove your tables from the database before exiting. Use the SQL DROP TABLE command followed by COMMIT to permanently delete the tables from the database.

```
DROP TABLE table-name;  
COMMIT;
```

These tables will be issued to you for the next assignment.

