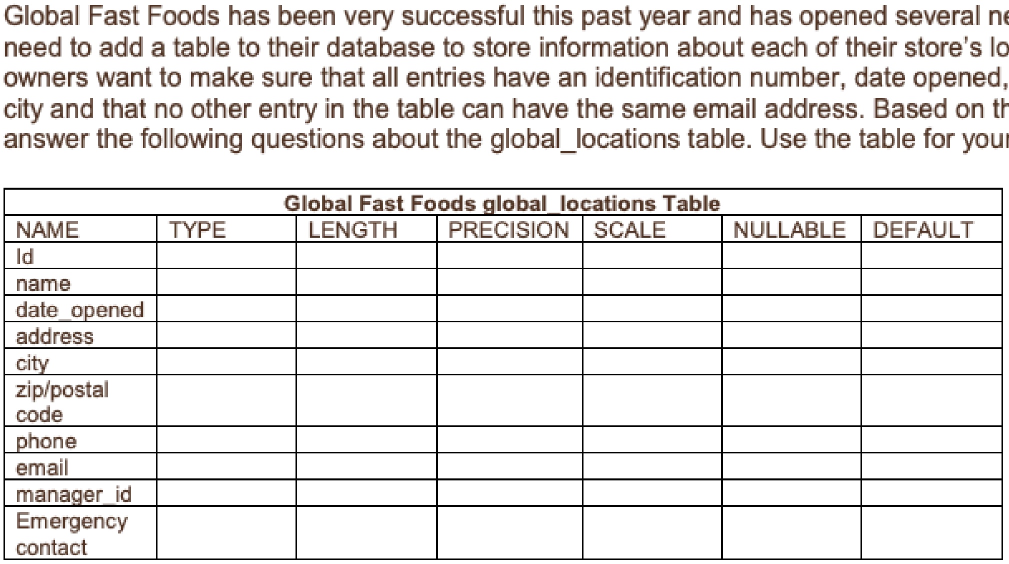
Name: Danny Chen
Assignment: Database Programming: Sections 12, 13
Date: 11/27/24

Table of Content

14-1 Intro to Constraints; NOT NULL and UNIQUE Constraints

Word	Definition
UNIQUE constraint	Every value in a column or set of columns (a composite key) must be unique
NOT NULL constraint	For every row entered into the table, there must be a value for that column
PRIMARY KEY	Constraint ensures that the column contains no null values and uniquely identifies each row of the table
CHECK constraint	Specifies a condition for a column that must be true for each row of data
REFERENCES	Identifies that table and column in the parent table
UNIQUE KEY	An integrity constraint that requires every value in a column or set of columns be unique
FOREIGN KEY	Designates a column (child table) that establishes a relationship between a primary key in the same table and a different table (parent table)
Table level constraint	References one or more columns and is defined separately from the definitions of the columns in the table

Constraint	Database rule.
Column level constraint	Database rule that references a single column



Global Fast Foods has been very successful this past year and has opened several new stores. They need to add a table to their database to store information about each of their store's locations. The owners want to make sure that all entries have an identification number, date opened, address, and city and that no other entry in the table can have the same email address. Based on this information, answer the following questions about the global_locations table. Use the table for your answers.

Global Fast Foods global_locations Table						
NAME	TYPE	LENGTH	PRECISION	SCALE	NULLABLE	DEFAULT
Id						
name						
date_opened						
address						
city						
zip/postal code						
phone						
email						
manager_id						
Emergency contact						

1. What is a "constraint" as it relates to data integrity?
    - a. **ANS:** "Constraint" enforce rules on data whenever it is modified (e.g. inserted, deleted, updated). It prevents a table or row(s) in the table from being deleted if it has dependencies from other tables.
2. What are the limitations of constraints that may be applied at the column level and at the table level?
    - a. **ANS:**
        - i. Constraints that can only be defined at column level
            1. NOT NULL constraint
        - ii. Can only be defined at table level
            1. Constraints that refer to more than 1 column (a composite key)
        - iii. Can be defined at column or table level
            1. UNIQUE, PRIMARY KEY, FOREIGN KEY, and CHECK constraints
        - iv. If CONSTRAINT is used in a CREATE TABLE statement, a name must be given to the constraint
3. Why is it important to give meaningful names to constraints?
    - a. **ANS:** It can be difficult to distinguish one constraint from another if the names are not meaningful (or too general). This especially will be the case as the database gets bigger and bigger.

4. Based on the information provided by the owners, choose a datatype for each column. Indicate the length, precision, and scale for each NUMBER datatype.
5. Use “nullable” to indicate those columns that can have null values.

ANS to 4 and 5:

Global Fast Foods global_locations Table						
NAME	TYPE	LENGTH	PRECISION	SCALE	NULLABLE	DEFAULT
Id	NUMBER	4	6	0		
name	VARCHAR2	30				
date opened	DATE		7			
address	VARCHAR2	50				
city	VARCHAR2	50				
zip/postal code	VARCHAR2	20				
phone	VARCHAR2	20			nullable	
email	VARCHAR2	100			nullable	
manager id	NUMBER	4	6	0		
Emergency contact	VARCHAR2	20			nullable	

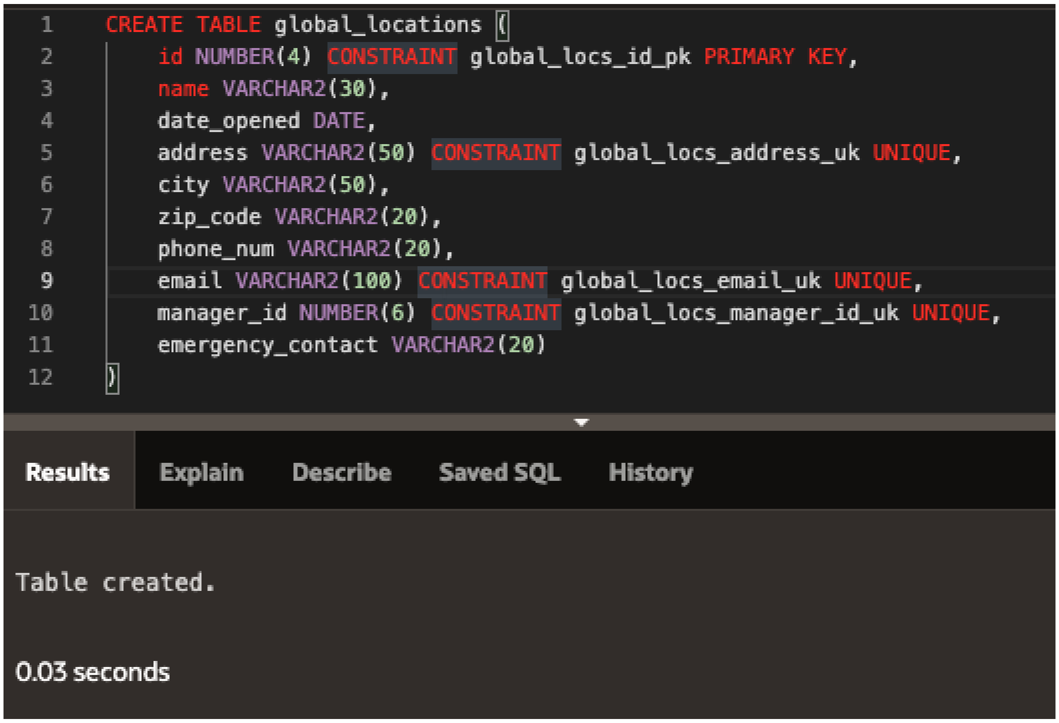
6. Write the CREATE TABLE statement for the Global Fast Foods locations table to define the constraints at the column level.

ANS:

```
CREATE TABLE global_locations (
   id NUMBER(4) CONSTRAINT global_locs_id_pk PRIMARY KEY,
   name VARCHAR2(30),
   date_opened DATE,
   address VARCHAR2(50) CONSTRAINT global_locs_address_uk UNIQUE,
   city VARCHAR2(50),
   zip_code VARCHAR2(20),
   phone_num VARCHAR2(20),
   email VARCHAR2(100) CONSTRAINT global_locs_email_uk UNIQUE,
   manager_id NUMBER(6) CONSTRAINT global_locs_manager_id_uk UNIQUE,
    emergency_contact VARCHAR2(20)
)
```

7. Execute the CREATE TABLE statement in Oracle Application Express.

ANS:



```
CREATE TABLE global_locations (
    id NUMBER(4) CONSTRAINT global_locs_id_pk PRIMARY KEY,
    name VARCHAR2(30),
    date_opened DATE,
    address VARCHAR2(50) CONSTRAINT global_locs_address_uk UNIQUE,
    city VARCHAR2(50),
    zip_code VARCHAR2(20),
    phone_num VARCHAR2(20),
    email VARCHAR2(100) CONSTRAINT global_locs_email_uk UNIQUE,
    manager_id NUMBER(6) CONSTRAINT global_locs_manager_id_uk UNIQUE,
    emergency_contact VARCHAR2(20)
)
```

```
Table created.

0.03 seconds
```

8. Execute a DESCRIBE command to view the Table Summary information.

ANS:

DESCRIBE global_locations

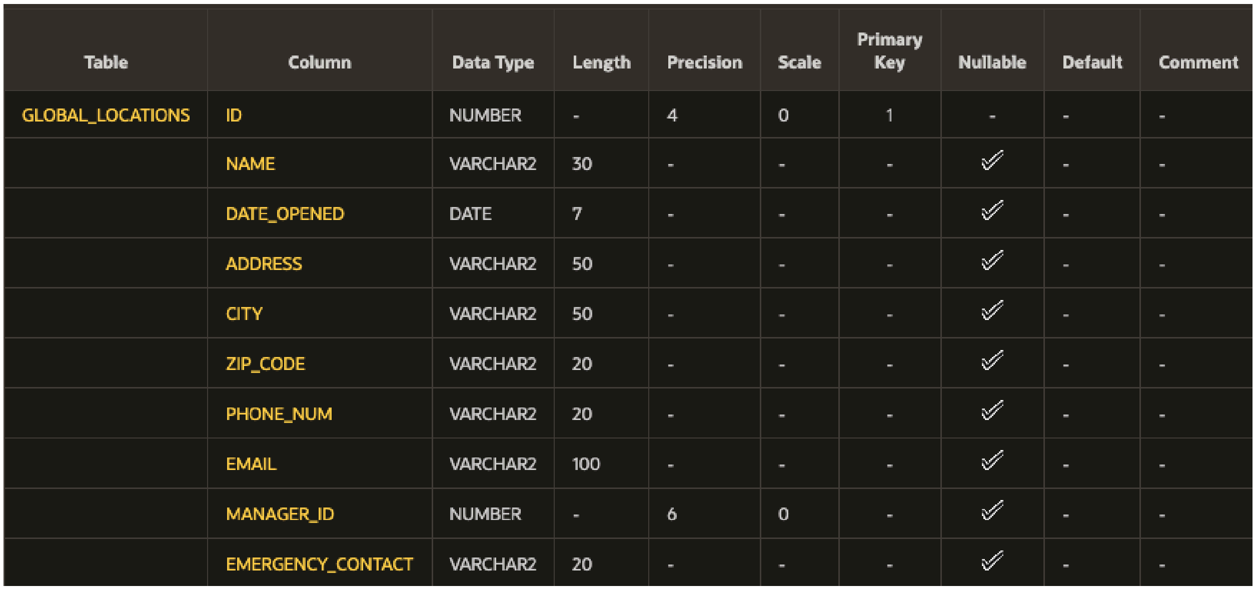


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
GLOBAL_LOCATIONS	ID	NUMBER	-	4	0	1	-	-	-
	NAME	VARCHAR2	30	-	-	-	✓	-	-
	DATE_OPENED	DATE	7	-	-	-	✓	-	-
	ADDRESS	VARCHAR2	50	-	-	-	✓	-	-
	CITY	VARCHAR2	50	-	-	-	✓	-	-
	ZIP_CODE	VARCHAR2	20	-	-	-	✓	-	-
	PHONE_NUM	VARCHAR2	20	-	-	-	✓	-	-
	EMAIL	VARCHAR2	100	-	-	-	✓	-	-
	MANAGER_ID	NUMBER	-	6	0	-	✓	-	-
	EMERGENCY_CONTACT	VARCHAR2	20	-	-	-	✓	-	-

9. Rewrite the CREATE TABLE statement for the Global Fast Foods locations table to define the UNIQUE constraints at the table level. Do not execute this statement.

ANS:

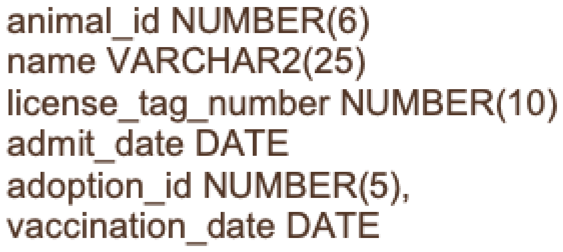
```
CREATE TABLE global_locations (
   id NUMBER(4),
   name VARCHAR2(30),
   date_opened DATE,
   address VARCHAR2(50),
   city VARCHAR2(50),
   zip_code VARCHAR2(20),
   phone_num VARCHAR2(20),
   email VARCHAR2(100),
   manager_id NUMBER(6),
   emergency_contact VARCHAR2(20),

   CONSTRAINT global_locs_id_pk PRIMARY KEY,
   CONSTRAINT global_locs_address_uk UNIQUE,
   CONSTRAINT global_locs_email_uk UNIQUE,
   CONSTRAINT global_locs_manager_id_uk UNIQUE,
)
```

14-2 PRIMARY KEY, FOREIGN KEY, and CHECK Constraints

Word	Definition
ON DELETE CASCADE	Allows a foreign key row that is referenced to a primary key row to be deleted
CHECK constraint	Explicitly defines a condition that must be met
PRIMARY KEY constraint	A column or set of columns that uniquely identifies each row in a table
NOT NULL constraint	Constraint ensures that the column contains no null values
ON DELETE SET NULL	Allows a child row to remain in a table with null values when a parent record has been deleted
FOREIGN KEY constraint	Establishes a relationship between the foreign key column and a primary key or unique key in the same table or a different table

1. What is the purpose of a
    a. PRIMARY KEY
        i. **ANS:** To help unique identify a row/record
    b. FOREIGN KEY (aka referential integrity constraints)
        i. **ANS:** Form relationships b/w tables
    c. CHECK CONSTRAINT
        i. **ANS:** Ensures the validity of data (e.g. making sure the end date is always greater than the start date b/c vice versa does not make sense)

2. Using the column information for the animals table below, name constraints where applicable at the table level, otherwise name them at the column level. Define the primary key (animal_id). The license_tag_number must be unique. The admit_date and vaccination_date columns cannot contain null values.

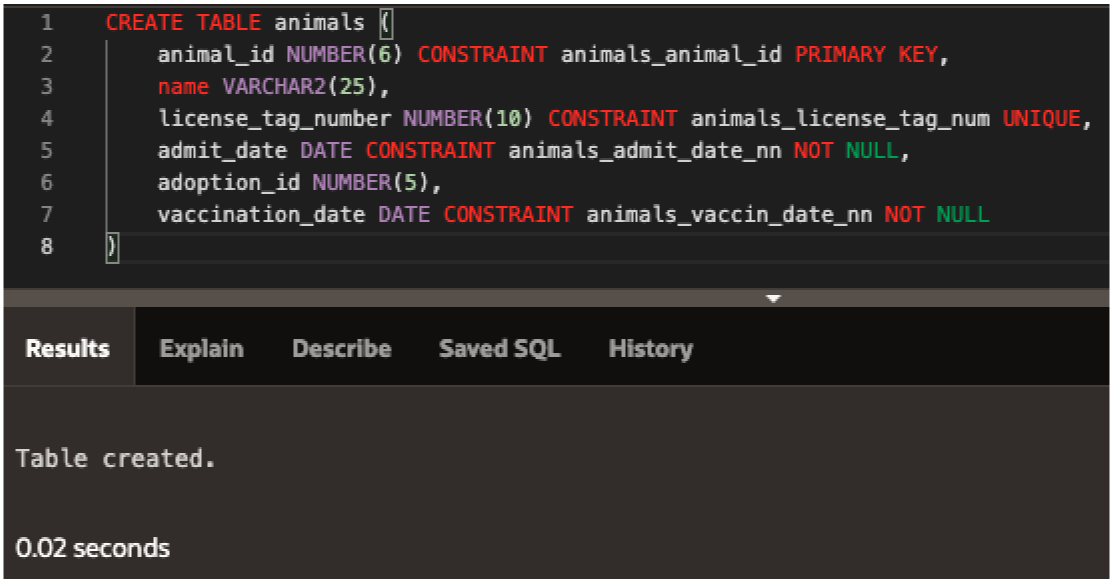


```
animal_id NUMBER(6)
name VARCHAR2(25)
license_tag_number NUMBER(10)
admit_date DATE
adoption_id NUMBER(5),
vaccination_date DATE
```

3. Create the animals table. Write the syntax you will use to create the table.

ANS for 2 and 3:

```
CREATE TABLE animals (
   animal_id NUMBER(6) CONSTRAINT animals_animal_id PRIMARY KEY,
   name VARCHAR2(25),
   license_tag_number NUMBER(10) CONSTRAINT animals_license_tag_num UNIQUE,
   admit_date DATE CONSTRAINT animals_admit_date_nn NOT NULL,
   adoption_id NUMBER(5),
   vaccination_date DATE CONSTRAINT animals_vaccin_date_nn NOT NULL
)
```



```
CREATE TABLE animals (
    animal_id NUMBER(6) CONSTRAINT animals_animal_id PRIMARY KEY,
    name VARCHAR2(25),
    license_tag_number NUMBER(10) CONSTRAINT animals_license_tag_num UNIQUE,
    admit_date DATE CONSTRAINT animals_admit_date_nn NOT NULL,
    adoption_id NUMBER(5),
    vaccination_date DATE CONSTRAINT animals_vaccin_date_nn NOT NULL
)
```

```
Table created.

0.02 seconds
```

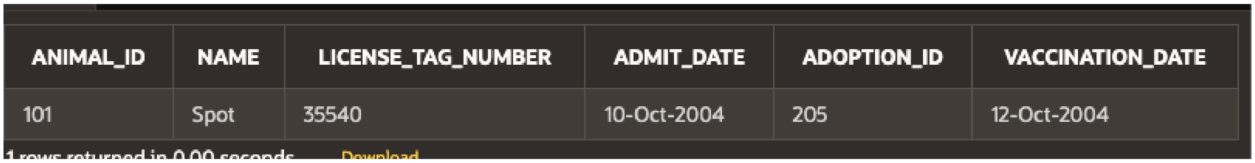
4. Enter one row into the table. Execute a SELECT * statement to verify your input. Refer to the graphic below for input.

ANIMAL_ ID	NAME	LICENSE_TAG_ NUMBER	ADMIT_DATE	ADOPTION_ ID	VACCINATION_ DATE
101	Spot	35540	10-Oct-2004	205	12-Oct-2004

ANS:

```
INSERT INTO animals (
   animal_id,
   name,
   license_tag_number,
   admit_date,
   adoption_id,
   vaccination_date
)
VALUES (101, 'Spot', 35540, '10-Oct-2004', 205, '12-Oct-2004')
```

SELECT * FROM animals



ANIMAL_ID	NAME	LICENSE_TAG_NUMBER	ADMIT_DATE	ADOPTION_ID	VACCINATION_DATE
101	Spot	35540	10-Oct-2004	205	12-Oct-2004

1 rows returned in 0.00 seconds   Download

5. Write the syntax to create a foreign key (adoption_id) in the animals table that has a corresponding primary- key reference in the adoptions table. Show both the column-level and table-level syntax. Note that because you have not actually created an adoptions table, no adoption_id primary key exists, so the foreign key cannot be added to the animals table.

ANS:

Column-level syntax

```
CREATE TABLE animals (
   animal_id NUMBER(6) CONSTRAINT animals_animal_id PRIMARY KEY,
   name VARCHAR2(25),
   license_tag_number NUMBER(10) CONSTRAINT animals_license_tag_num UNIQUE,
   admit_date DATE CONSTRAINT animals_admit_date_nn NOT NULL,
   adoption_id NUMBER(5) CONSTRAINT animals_adopt_id_fk REFERENCES adoptions(id),
   vaccination_date DATE CONSTRAINT animals_vaccin_date_nn NOT NULL
)
```

Table-level syntax

```
CREATE TABLE animals (
   animal_id NUMBER(6) CONSTRAINT animals_animal_id PRIMARY KEY,
   name VARCHAR2(25),
   license_tag_number NUMBER(10) CONSTRAINT animals_license_tag_num UNIQUE,
   admit_date DATE CONSTRAINT animals_admit_date_nn NOT NULL,
   adoption_id NUMBER(5),
   vaccination_date DATE CONSTRAINT animals_vaccin_date_nn NOT NULL

   CONSTRAINT animals_adopt_id_fk REFERENCES adoptions(id)
)
```

6. What is the effect of setting the foreign key in the ANIMAL table as:
   a. ON DELETE CASCADE

        i. **ANS:** When deleting a row from the adoptions table (parent table), any rows from the animals table (child table) that were linked to that row through a foreign key will be deleted too.

    b. ON DELETE SET NULL

        i. **ANS:** When deleting a row from the adoptions table (parent table), any rows from the animals table (child table) that were linked to that row through a foreign key will be filled with NULL rather than being deleted.

7. What are the restrictions on defining a CHECK constraint?
    a. The CHECK constraint's condition can refer to any column in the specified table, but not to columns of other tables.

14-3 Managing Constraints

Word	Definition
DISABLE CONSTRAINT	To deactivate an integrity constraint
CASCADE clause	Disables dependent integrity constraints
ALTER TABLE	To add, modify, or drop columns from a table
ENABLE CONSTRAINT	To activate an integrity constraint currently disabled
DROP CONSTRAINT	Removes a constraint from a table
DROP COLUMN	Allows user to delete a column from a table
CASCADE CONSTRAINT	Defines the actions the database server takes when a user attempts to delete or update a key to which existing foreign keys point

Using Oracle Application Express, click the SQL Workshop tab in the menu bar. Click the Object Browser and verify that you have a table named copy_d_clients and a table named copy_d_events. If you don't have these tables in your schema, create them before completing the exercises below. Here is how the original tables are related. The d_clients table has a primary key client_number. This has a primary-key constraint and it is referenced in the foreign-key constraint on the d_events table.

1. What are four functions that an ALTER statement can perform on constraints?
    a. **ANS:** ADD, DROP, ENABLE, and DISABLE

2. Since the tables are copies of the original tables, the integrity rules are not passed onto the new tables; only the column datatype definitions remain. You will need to add a PRIMARY KEY constraint to the copy_d_clients table. Name the primary key

copy_d_clients_pk. What is the syntax you used to create the PRIMARY KEY constraint to the copy_d_clients table?

ANS:

ALTER TABLE copy_d_clients
ADD CONSTRAINT copy_d_clients_pk PRIMARY KEY (client_number)

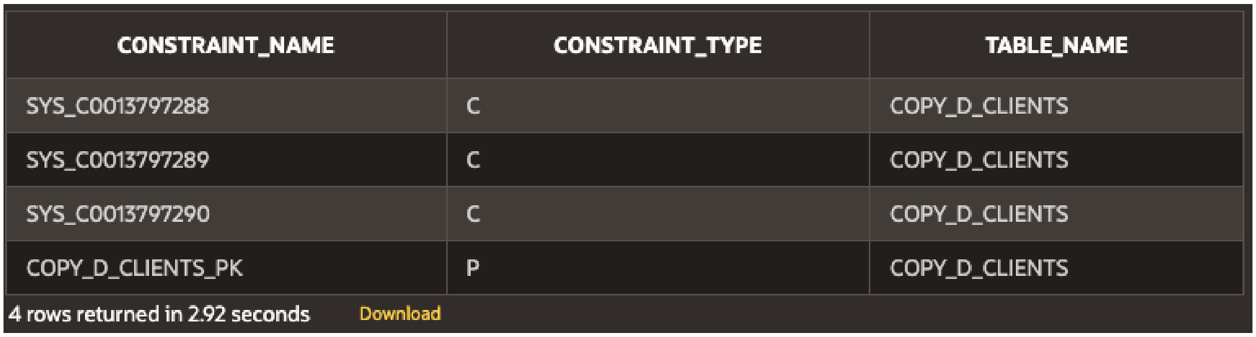
3. Create a FOREIGN KEY constraint in the copy_d_events table. Name the foreign key copy_d_events_fk. This key references the copy_d_clients table client_number column. What is the syntax you used to create the FOREIGN KEY constraint in the copy_d_events table?

ANS:

ALTER TABLE copy_d_events
ADD CONSTRAINT copy_d_events_fk FOREIGN KEY (client_number)
  REFERENCES copy_d_clients (client_number);

4. Use a SELECT statement to verify the constraint names for each of the tables. **Note that the table names must be *capitalized*.**
   a. The constraint name for the primary key in the copy_d_clients table is _______**COPY_D_CLIENTS_PK**________.

      QUERY USED:
      SELECT constraint_name, constraint_type, table_name
      FROM user_constraints
      WHERE table_name = UPPER('copy_d_clients');



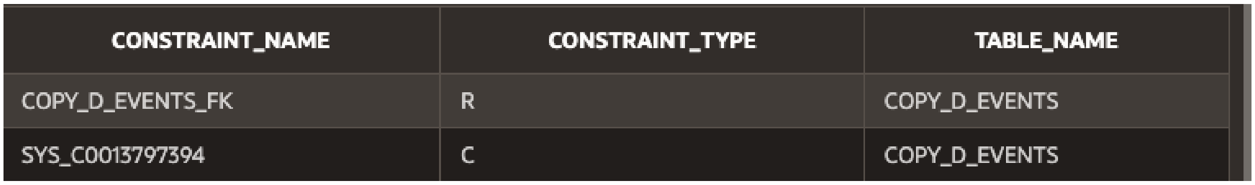
CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
SYS_C0013797288	C	COPY_D_CLIENTS
SYS_C0013797289	C	COPY_D_CLIENTS
SYS_C0013797290	C	COPY_D_CLIENTS
COPY_D_CLIENTS_PK	P	COPY_D_CLIENTS

4 rows returned in 2.92 seconds Download

   b. The constraint name for the foreign key in the copy_d_events table is _______**COPY_D_EVENTS_FK**________.

      QUERY USED:
      SELECT constraint_name, constraint_type, table_name

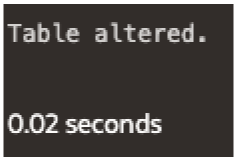
FROM user_constraints
WHERE table_name = UPPER('copy_d_events');



CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
COPY_D_EVENTS_FK	R	COPY_D_EVENTS
SYS_C0013797394	C	COPY_D_EVENTS

5. Drop the PRIMARY KEY constraint on the copy_d_clients table. Explain your results.
    a. **NOTE:** Don't put quotes around the constraint name or else there's errors

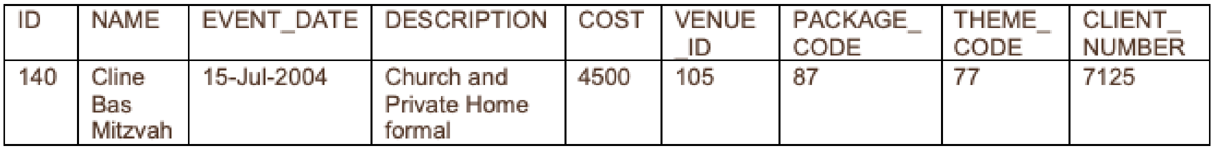
ANS:
ALTER TABLE copy_d_clients
DROP CONSTRAINT copy_d_clients_pk CASCADE;



```
Table altered.

0.02 seconds
```

Results
The constraint name "copy_d_clients_pk" has been removed from USER_CONSTRAINT where table name is UPPER('copy_d_clients'). Since I used CASCADE, the foreign key from USER_CONSTRAINT where table name is UPPER('copy_d_events') has also been removed. CASCADE causes dependent constraints to be dropped too.

6. Add the following event to the copy_d_events table. Explain your results.

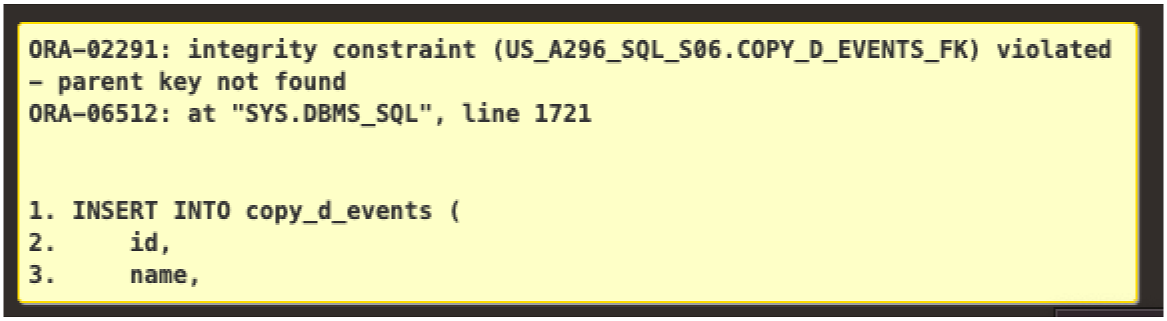


ID	NAME	EVENT_DATE	DESCRIPTION	COST	VENUE_ID	PACKAGE_CODE	THEME_CODE	CLIENT_NUMBER
140	Cline Bas Mitzvah	15-Jul-2004	Church and Private Home formal	4500	105	87	77	7125

ANS:
NOTE: The PRIMARY KEY constraint was added back to the copy_d_events table before applying the query below

INSERT INTO copy_d_events (
    id,
    name,
    event_date,
    description,
    cost,

```
    venue_id,
    package_code,
    theme_code,
    client_number
)
VALUES (
    140,
    'Cline Bas Mitzvah',
    '15-Jul-2004',
    'Church and Private Home formal',
    4500,
    105,
    87,
    77,
    7125
)
```



```
ORA-02291: integrity constraint (US_A296_SQL_S06.COPY_D_EVENTS_FK) violated
- parent key not found
ORA-06512: at "SYS.DBMS_SQL", line 1721


INSERT INTO copy_d_events (
    id,
    name,
```

Results
Since client number 7125 does not exist in the parent table (copy_d_clients), trying to insert a row with that client number into copy_d_events would not work since the child table's (copy_d_events) client number column is a foreign key that references the parent table's (copy_d_clients) client number

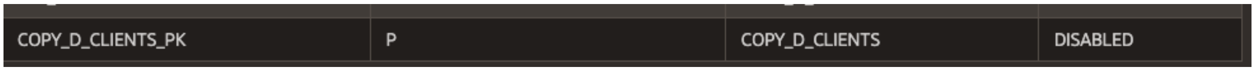
7. Create an ALTER TABLE query to disable the primary key in the copy_d_clients table. Then add the values from #6 to the copy_d_events table. Explain your results.

ANS:

```
ALTER TABLE copy_d_clients
DISABLE CONSTRAINT copy_d_clients_pk CASCADE;
```

Results
Running the below query shows that the primary key of copy_d_client table has been disabled:

```
SELECT constraint_name, constraint_type, table_name, status
FROM user_constraints
WHERE table_name = UPPER('copy_d_clients');
```



COPY_D_CLIENTS_PK	P	COPY_D_CLIENTS	DISABLED

When inserting the values from #6 into the copy_d_events table, the operation completes successfully. However, client number 7125 in the new row does not match any client number in the copy_d_clients table. As a result, client number 7125 does not reference any client number of a record in the copy_d_clients table, despite being designated as a foreign key.

8. Repeat question 6: Insert the new values in the copy_d_events table. Explain your results.

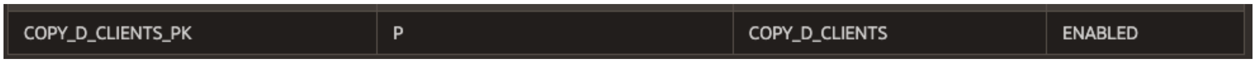
ANS:
The operation still completed successfully. Since the primary key was disabled in the copy_d_clients table, the rows are no longer required to have values in one column or a combination of columns that uniquely identify them.

9. Enable the primary-key constraint in the copy_d_clients table. Explain your results.

ANS:

```
ALTER TABLE copy_d_clients
ENABLE CONSTRAINT copy_d_clients_pk;
```

Results

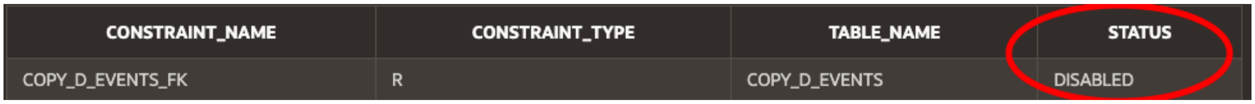


COPY_D_CLIENTS_PK	P	COPY_D_CLIENTS	ENABLED

10. If you wanted to enable the foreign-key column and reestablish the referential integrity between these two tables, what must be done?

ANS:
Since I used CASCADE to disable the primary key, that also disabled dependent integrity constraints. Even if I enable the primary key constraint now, the dependent constraints are not automatically enabled as shown by running query below:

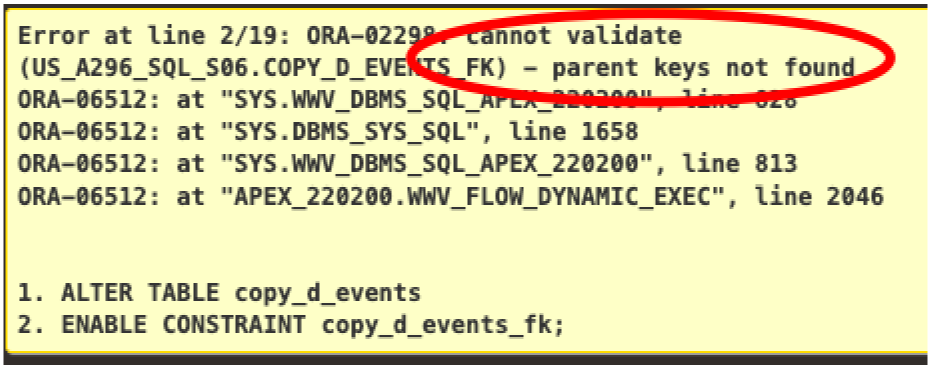
```
SELECT constraint_name, constraint_type, table_name, status
FROM user_constraints
WHERE table_name = UPPER('copy_d_events');
```



CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME	STATUS
COPY_D_EVENTS_FK	R	COPY_D_EVENTS	DISABLED

Running the below query to enable the foreign key would result in the following error:

ALTER TABLE copy_d_events
ENABLE CONSTRAINT copy_d_events_fk;



```
Error at line 2/19: ORA-02298: cannot validate
(US_A296_SQL_S06.COPY_D_EVENTS_FK) - parent keys not found
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_220200", line 828
ORA-06512: at "SYS.DBMS_SYS_SQL", line 1658
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_220200", line 813
ORA-06512: at "APEX_220200.WWV_FLOW_DYNAMIC_EXEC", line 2046


1. ALTER TABLE copy_d_events
2. ENABLE CONSTRAINT copy_d_events_fk;
```

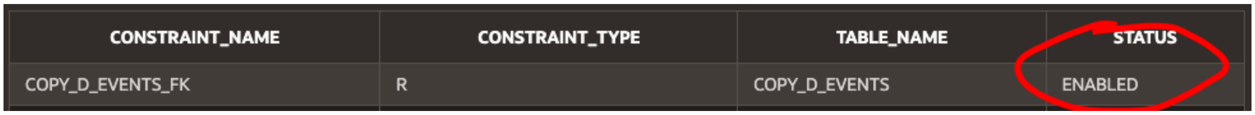
<u>To re-enable foreign-key column and reestablish the referential integrity b/w the two tables:</u>

1. Remove the rows in the child table whose client number (foreign key) does not match any client number in the parent table.

```
DELETE FROM copy_d_events
WHERE client_number NOT IN (
   SELECT client_number
   FROM copy_d_clients
);
```

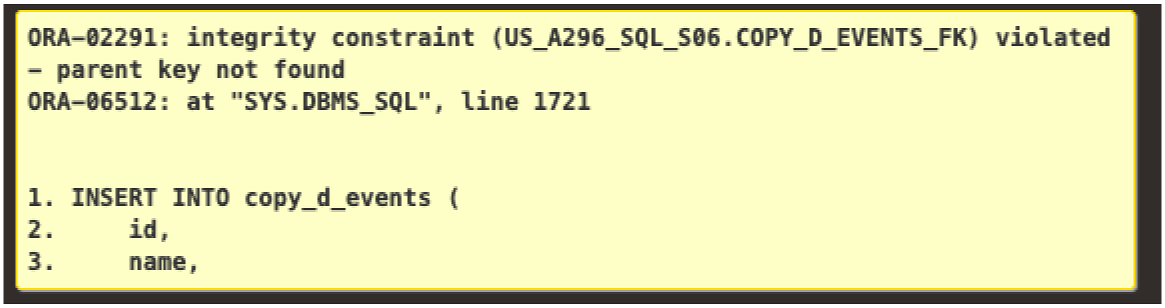
2. There are no more rows in the child table that do not match parent keys. Now the below query will run successfully.

ALTER TABLE copy_d_events
ENABLE CONSTRAINT copy_d_events_fk;



CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME	STATUS
COPY_D_EVENTS_FK	R	COPY_D_EVENTS	ENABLED

3. Attempting to insert the values from #6 will result in the error below. The integrity constraint is violated since a parent key is not found.

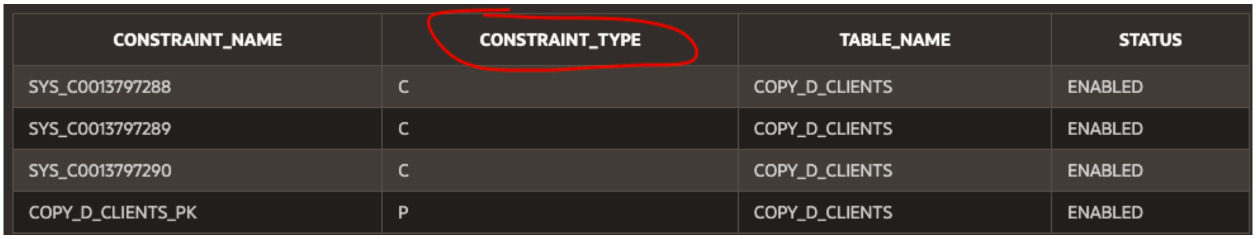


```
ORA-02291: integrity constraint (US_A296_SQL_S06.COPY_D_EVENTS_FK) violated
- parent key not found
ORA-06512: at "SYS.DBMS_SQL", line 1721


INSERT INTO copy_d_events (
    id,
    name,
```

11. Why might you want to disable and then re-enable a constraint?
    a. **ANS:**
        i. To load large amounts of data into a table quicker
        ii. When performing operations in bulk that make massive changes to a table (e.g. changing everyone's employee number by adding 1,000 to the existing number)

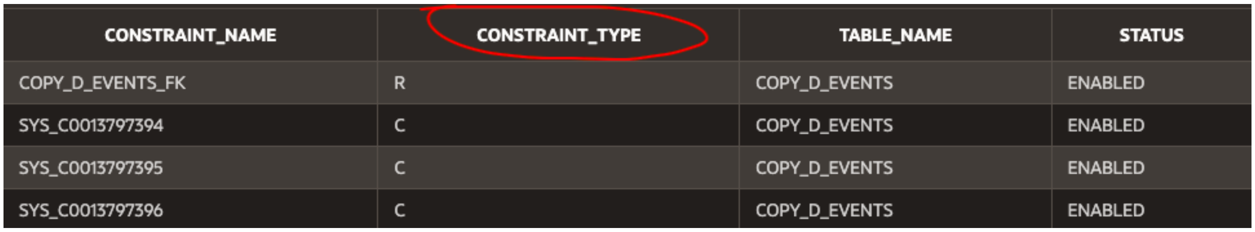
12. Query the data dictionary for some of the constraints that you have created. How does the data dictionary identify each constraint type?
    a. **ANS:**
        i. R = REFERENCES (foreign key)
        ii. P = PRIMARY KEY
        iii. C = CHECK constraint (including NOT NULL);

SELECT constraint_name, constraint_type, table_name, status
FROM user_constraints
WHERE table_name = UPPER('copy_d_clients');



CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME	STATUS
SYS_C0013797288	C	COPY_D_CLIENTS	ENABLED
SYS_C0013797289	C	COPY_D_CLIENTS	ENABLED
SYS_C0013797290	C	COPY_D_CLIENTS	ENABLED
COPY_D_CLIENTS_PK	P	COPY_D_CLIENTS	ENABLED

SELECT constraint_name, constraint_type, table_name, status
FROM user_constraints
WHERE table_name = UPPER('copy_d_events');



CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME	STATUS
COPY_D_EVENTS_FK	R	COPY_D_EVENTS	ENABLED
SYS_C0013797394	C	COPY_D_EVENTS	ENABLED
SYS_C0013797395	C	COPY_D_EVENTS	ENABLED
SYS_C0013797396	C	COPY_D_EVENTS	ENABLED

15-1 Creating Views

Word	Definition
View	A subset of data from one or more tables that is generated from a query and stored as a virtual table
VIEW_NAME	Name of view
FORCE	Creates a view regardless of whether or not the base tables exist
Simple view	Derives data from a table, no functions or groups, performs DML operations through the view
NOFORCE	Creates the view only if the base table exists
CREATE VIEW	Statement used to create a new view
Alias	Specifies a name for each expression selected by the view's query
Subquery	A complete SELECT statement
Complex view	Derives data from more than one table, contains functions or groups of data, and does not always allow DML operations through the view
REPLACE	Re-creates the view if it already exists

1. What are three uses for a view from a DBA’s perspective?
    a. **ANS:**
        i. Allow users to interact with the database using simple queries (in other words, reduce the complexity of executing queries that are more complicated)
        ii. The DBA can provides groups of users with access to data and set their own limits on what they can do
        iii. Views can obtain data from several tables, which provides data independence for users

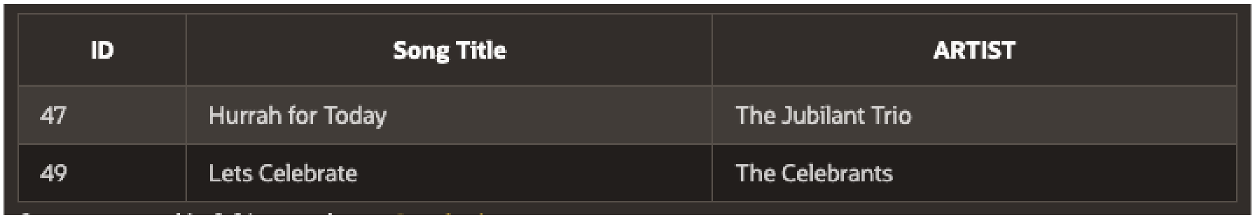
2. Create a simple view called view_d_songs that contains the ID, title, and artist from the DJs on Demand table for each “New Age” type code. In the subquery, use the alias “Song Title” for the title column.

ANS:

```
CREATE VIEW view_d_songs
AS SELECT id, title  "Song Title", artist
   FROM d_songs
   WHERE type_code = (
      SELECT code
      FROM d_types
      WHERE LOWER(description) = 'new age'
   )
```

3. SELECT * FROM view_d_songs. What was returned?

ANS:



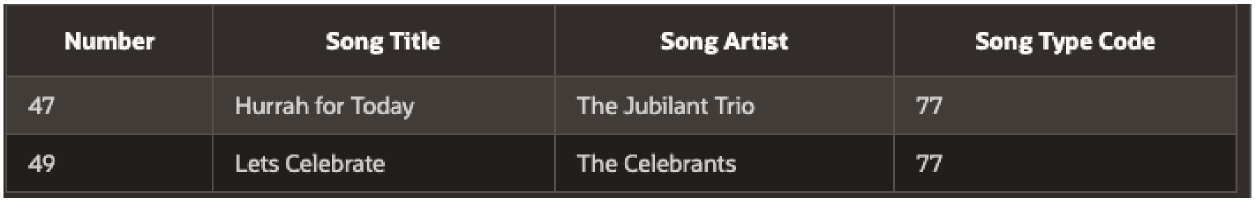
ID	Song Title	ARTIST
47	Hurrah for Today	The Jubilant Trio
49	Lets Celebrate	The Celebrants

4. REPLACE view_d_songs. Add type_code to the column list. Use aliases for all columns.

ANS:

NOTE: Only one query should be run at a time. Run the query to create (or replace) the view first, then delete it and paste the SELECT statement in. Running both together leads to an error.

```
CREATE OR REPLACE VIEW view_d_songs
AS SELECT id "Number", title  "Song Title", artist "Song Artist", type_code "Song Type Code"
   FROM d_songs
   WHERE type_code = (
      SELECT code
      FROM d_types
      WHERE LOWER(description) = 'new age'
   )

SELECT * FROM view_d_songs.
```



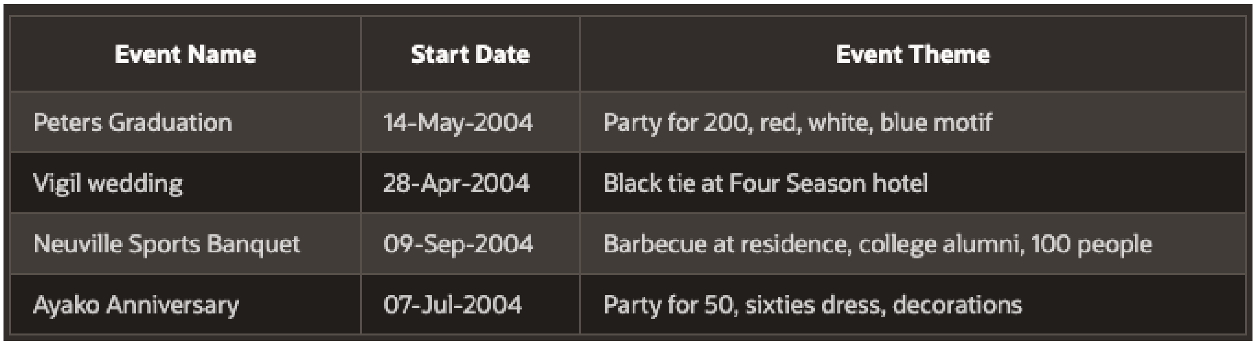
Number	Song Title	Song Artist	Song Type Code
47	Hurrah for Today	The Jubilant Trio	77
49	Lets Celebrate	The Celebrants	77

5. Jason Tsang, the disk jockey for DJs on Demand, needs a list of the past events and those planned for the coming months so he can make arrangements for each event's equipment setup. As the company manager, you do not want him to have access to the price that clients paid for their events. Create a view for Jason to use that displays the name of the event, the event date, and the theme description. Use aliases for each column name.

ANS:

CREATE VIEW view_d_events
AS SELECT name "Event Name", event_date "Start Date", description "Event Theme"
  FROM copy_d_events

SELECT * FROM view_d_events;



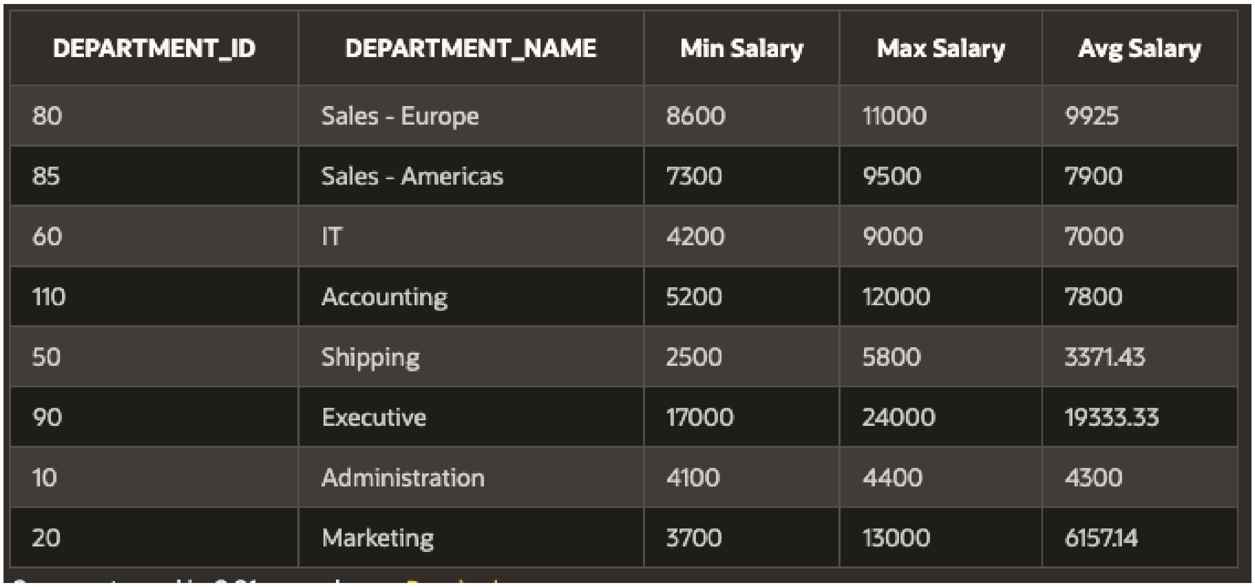
Event Name	Start Date	Event Theme
Peters Graduation	14-May-2004	Party for 200, red, white, blue motif
Vigil wedding	28-Apr-2004	Black tie at Four Season hotel
Neuville Sports Banquet	09-Sep-2004	Barbecue at residence, college alumni, 100 people
Ayako Anniversary	07-Jul-2004	Party for 50, sixties dress, decorations

6. It is company policy that only upper-level management be allowed access to individual employee salaries. The department managers, however, need to know the minimum, maximum, and average salaries, grouped by department. Use the Oracle database to prepare a view that displays the needed information for department managers.

ANS:

CREATE VIEW view_dept_emp_salary
AS SELECT e.department_id, d.department_name, MIN(e.salary) "Min Salary", MAX(e.salary) "Max Salary", ROUND(AVG(e.salary), 2) "Avg Salary"
  FROM employees e JOIN departments d
  ON e.department_id = d.department_id
  GROUP BY e.department_id, d.department_name

SELECT * FROM view_dept_emp_salary



DEPARTMENT_ID	DEPARTMENT_NAME	Min Salary	Max Salary	Avg Salary
80	Sales - Europe	8600	11000	9925
85	Sales - Americas	7300	9500	7900
60	IT	4200	9000	7000
110	Accounting	5200	12000	7800
50	Shipping	2500	5800	3371.43
90	Executive	17000	24000	19333.33
10	Administration	4100	4400	4300
20	Marketing	3700	13000	6157.14

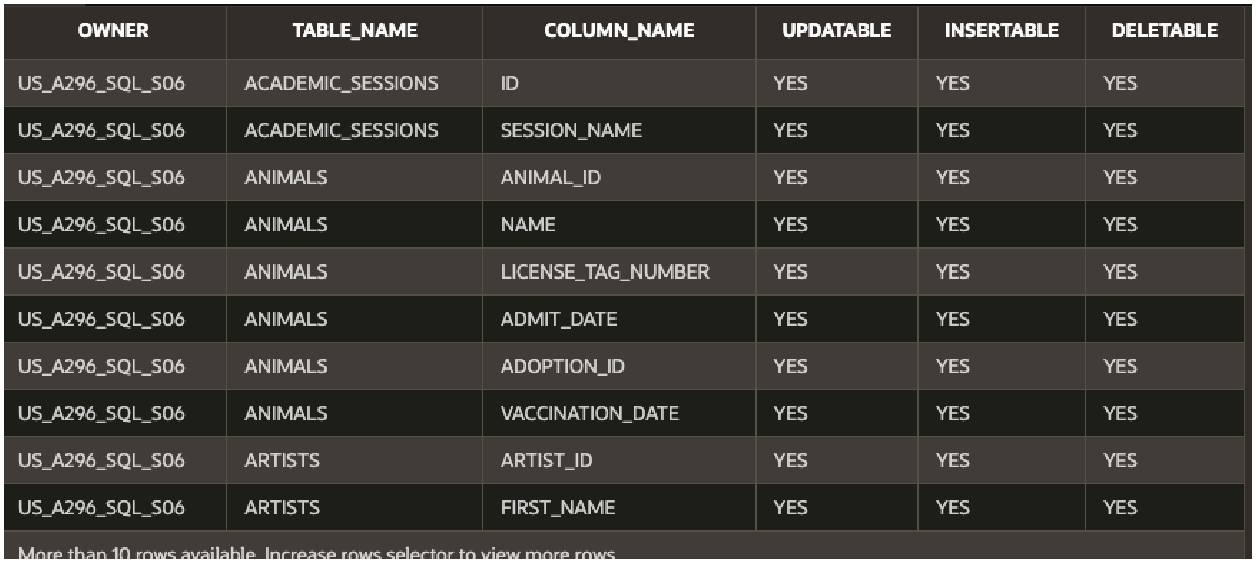
15-2 DML Operations and Views

Word	Definition
ROWNUM	A pseudocolumn which assigns a sequential value starting with 1 to each of the rows returned from the subquery
WITH CHECK OPTION	Specifies that INSERTS and UPDATES performed through the view can't create rows which the view cannot select
WITH READ ONLY	Ensures that no DML operations can be performed on this view

1. Query the data dictionary USER_UPDATABLE_COLUMNS to make sure the columns in the base tables will allow UPDATE, INSERT, or DELETE. Use a SELECT statement. All table names in the data dictionary are stored in uppercase.

ANS:

```
SELECT *
FROM USER_UPDATABLE_COLUMNS
```



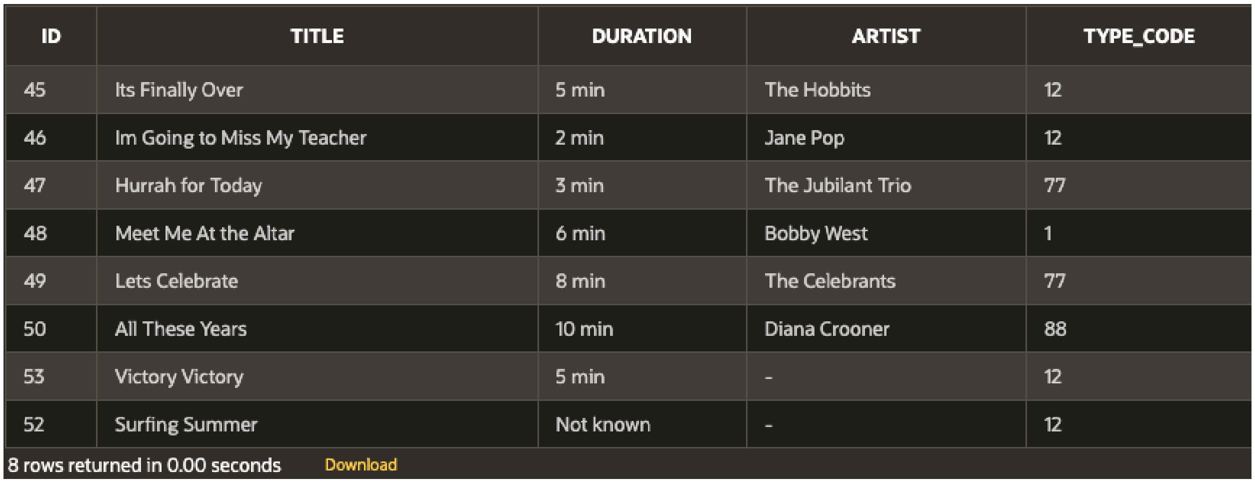
OWNER	TABLE_NAME	COLUMN_NAME	UPDATABLE	INSERTABLE	DELETABLE
US_A296_SQL_S06	ACADEMIC_SESSIONS	ID	YES	YES	YES
US_A296_SQL_S06	ACADEMIC_SESSIONS	SESSION_NAME	YES	YES	YES
US_A296_SQL_S06	ANIMALS	ANIMAL_ID	YES	YES	YES
US_A296_SQL_S06	ANIMALS	NAME	YES	YES	YES
US_A296_SQL_S06	ANIMALS	LICENSE_TAG_NUMBER	YES	YES	YES
US_A296_SQL_S06	ANIMALS	ADMIT_DATE	YES	YES	YES
US_A296_SQL_S06	ANIMALS	ADOPTION_ID	YES	YES	YES
US_A296_SQL_S06	ANIMALS	VACCINATION_DATE	YES	YES	YES
US_A296_SQL_S06	ARTISTS	ARTIST_ID	YES	YES	YES
US_A296_SQL_S06	ARTISTS	FIRST_NAME	YES	YES	YES

More than 10 rows available. Increase rows selector to view more rows.

2. Use the CREATE or REPLACE option to create a view of all the columns in the copy_d_songs table called view_copy_d_songs.

ANS:

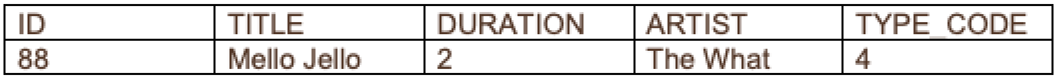
```
CREATE OR REPLACE VIEW view_copy_d_songs
AS SELECT *
  FROM copy_d_songs

SELECT * FROM view_copy_d_songs;
```



ID	TITLE	DURATION	ARTIST	TYPE_CODE
45	Its Finally Over	5 min	The Hobbits	12
46	Im Going to Miss My Teacher	2 min	Jane Pop	12
47	Hurrah for Today	3 min	The Jubilant Trio	77
48	Meet Me At the Altar	6 min	Bobby West	1
49	Lets Celebrate	8 min	The Celebrants	77
50	All These Years	10 min	Diana Crooner	88
53	Victory Victory	5 min	-	12
52	Surfing Summer	Not known	-	12

8 rows returned in 0.00 seconds Download

3. Use view_copy_d_songs to INSERT the following data into the underlying copy_d_songs table. Execute a SELECT * from copy_d_songs to verify your DML command. See the graphic.



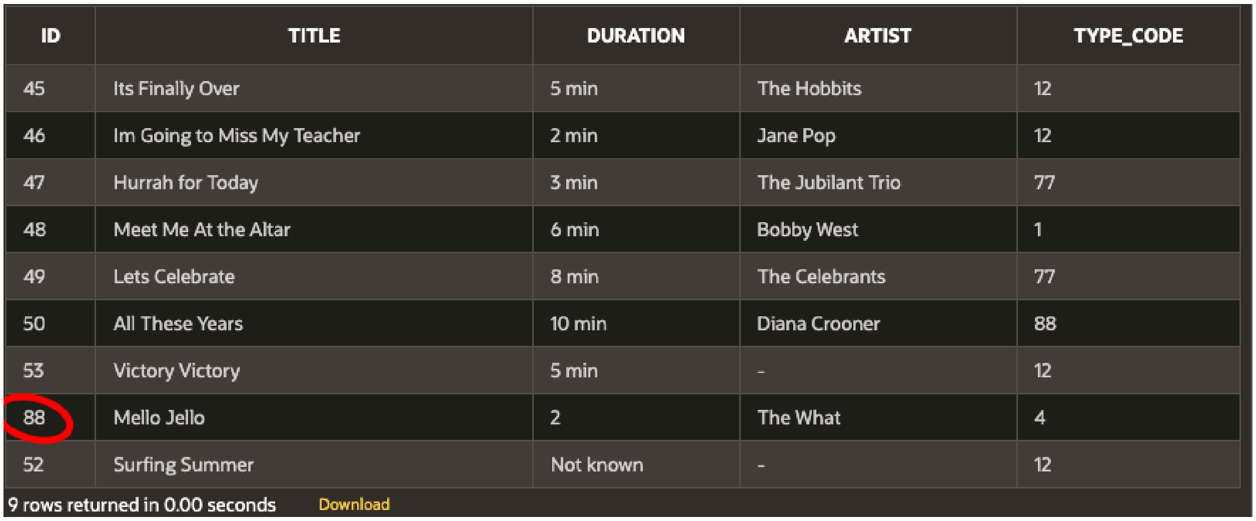
ID	TITLE	DURATION	ARTIST	TYPE_CODE
88	Mello Jello	2	The What	4

ANS:

```
INSERT INTO view_copy_d_songs
   (id, title, duration, artist, type_code)
VALUES
   (88, 'Mello Jello', 2, 'The What', 4)
```

<u>Both queries below return tables that include the newly inserted row</u>

```
SELECT * FROM copy_d_songs
SELECT * FROM view_copy_d_songs;
```



ID	TITLE	DURATION	ARTIST	TYPE_CODE
45	Its Finally Over	5 min	The Hobbits	12
46	Im Going to Miss My Teacher	2 min	Jane Pop	12
47	Hurrah for Today	3 min	The Jubilant Trio	77
48	Meet Me At the Altar	6 min	Bobby West	1
49	Lets Celebrate	8 min	The Celebrants	77
50	All These Years	10 min	Diana Crooner	88
53	Victory Victory	5 min	-	12
88	Mello Jello	2	The What	4
52	Surfing Summer	Not known	-	12

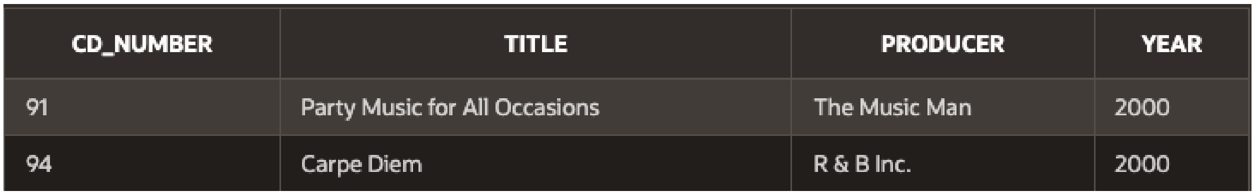
9 rows returned in 0.00 seconds Download

4. Create a view based on the DJs on Demand `COPY_D_CDS` table. Name the view `read_copy_d_cds`. Select all columns to be included in the view. Add a WHERE clause to restrict the year to 2000. Add the `WITH READ ONLY` option.

ANS:

```
CREATE OR REPLACE VIEW read_copy_d_cds
AS SELECT *
  FROM copy_d_cds
  WHERE year = '2000'
WITH READ ONLY;
```

```
SELECT * FROM read_copy_d_cds
```



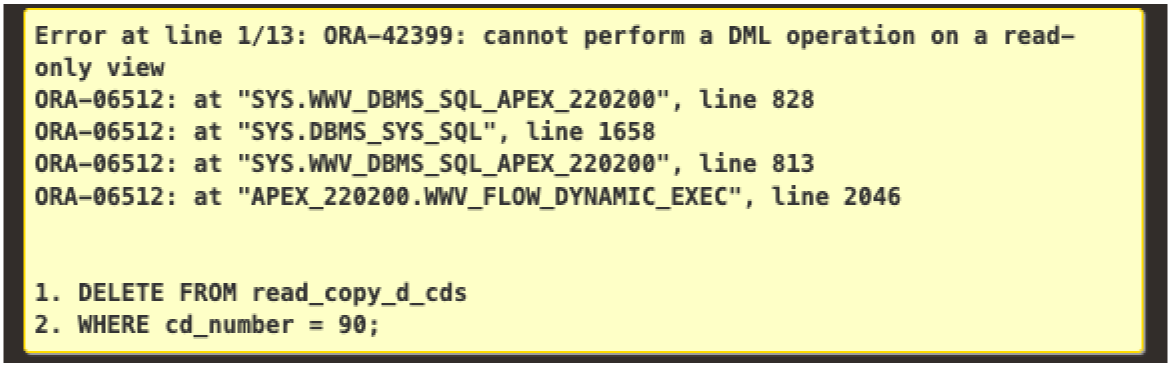
CD_NUMBER	TITLE	PRODUCER	YEAR
91	Party Music for All Occasions	The Music Man	2000
94	Carpe Diem	R & B Inc.	2000

5. Using the `read_copy_d_cds` view, execute a `DELETE FROM read_copy_d_cds WHERE cd_number = 90;`

ANS:

DELETE FROM read_copy_d_cds
WHERE cd_number = 90;

Since the view is read-only, the error "cannot perform a DML operation on a read-only view" pops up

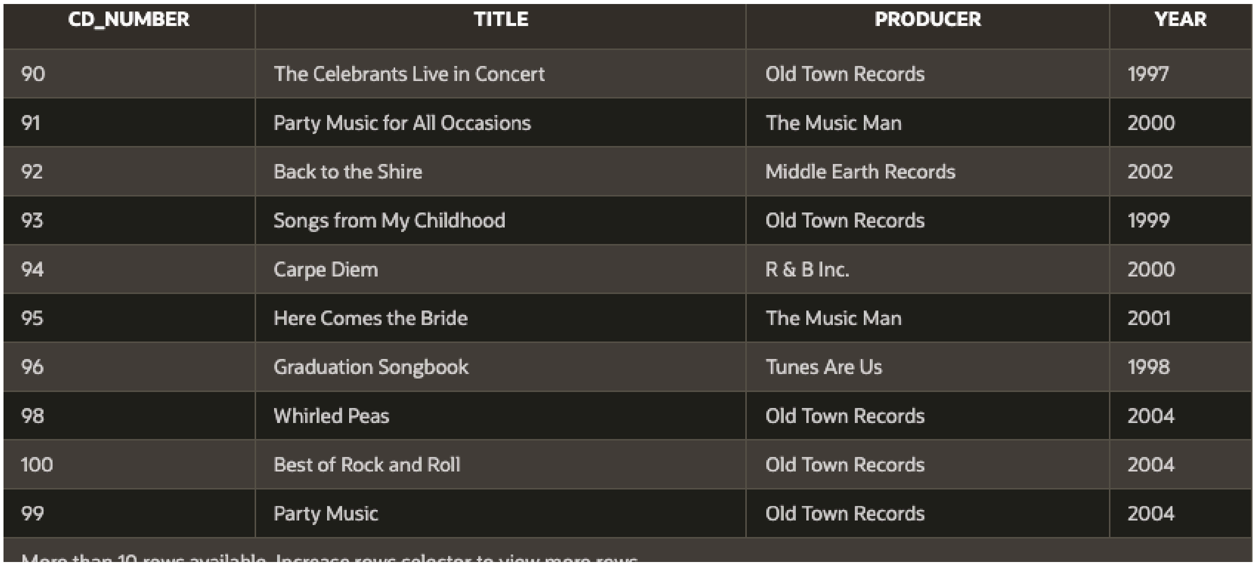


```
Error at line 1/13: ORA-42399: cannot perform a DML operation on a read-
only view
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_220200", line 828
ORA-06512: at "SYS.DBMS_SYS_SQL", line 1658
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_220200", line 813
ORA-06512: at "APEX_220200.WWV_FLOW_DYNAMIC_EXEC", line 2046


1. DELETE FROM read_copy_d_cds
2. WHERE cd_number = 90;
```

6. Use `REPLACE` to modify `read_copy_d_cds`. Replace the `WITH READ ONLY` option with `WITH CHECK OPTION CONSTRAINT ck_read_copy_d_cds`. Execute a `SELECT *` statement to verify that the view exists.

ANS:

```
CREATE OR REPLACE VIEW read_copy_d_cds
AS SELECT *
  FROM copy_d_cds
WITH CHECK OPTION CONSTRAINT ck_read_copy_d_cds;
```



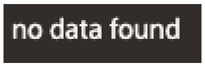
CD_NUMBER	TITLE	PRODUCER	YEAR
90	The Celebrants Live in Concert	Old Town Records	1997
91	Party Music for All Occasions	The Music Man	2000
92	Back to the Shire	Middle Earth Records	2002
93	Songs from My Childhood	Old Town Records	1999
94	Carpe Diem	R & B Inc.	2000
95	Here Comes the Bride	The Music Man	2001
96	Graduation Songbook	Tunes Are Us	1998
98	Whirled Peas	Old Town Records	2004
100	Best of Rock and Roll	Old Town Records	2004
99	Party Music	Old Town Records	2004

More than 10 rows available. Increase rows selector to view more rows.

7. Use the `read_copy_d_cds` view to delete any CD of year 2000 from the underlying `copy_d_cds`.

ANS:

```
DELETE FROM read_copy_d_cds
WHERE year = '2000'

SELECT * FROM read_copy_d_cds
WHERE year = '2000';
```

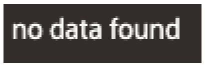


```
no data found
```

8. Use the `read_copy_d_cds` view to delete `cd_number = 90` from the underlying `copy_d_cds` table.

ANS:

```
DELETE FROM read_copy_d_cds
WHERE cd_number = 90;

SELECT * FROM read_copy_d_cds
WHERE cd_number = 90;
```



```
no data found
```

9. Use the `read_copy_d_cds` view to delete year 2001 records.

ANS:

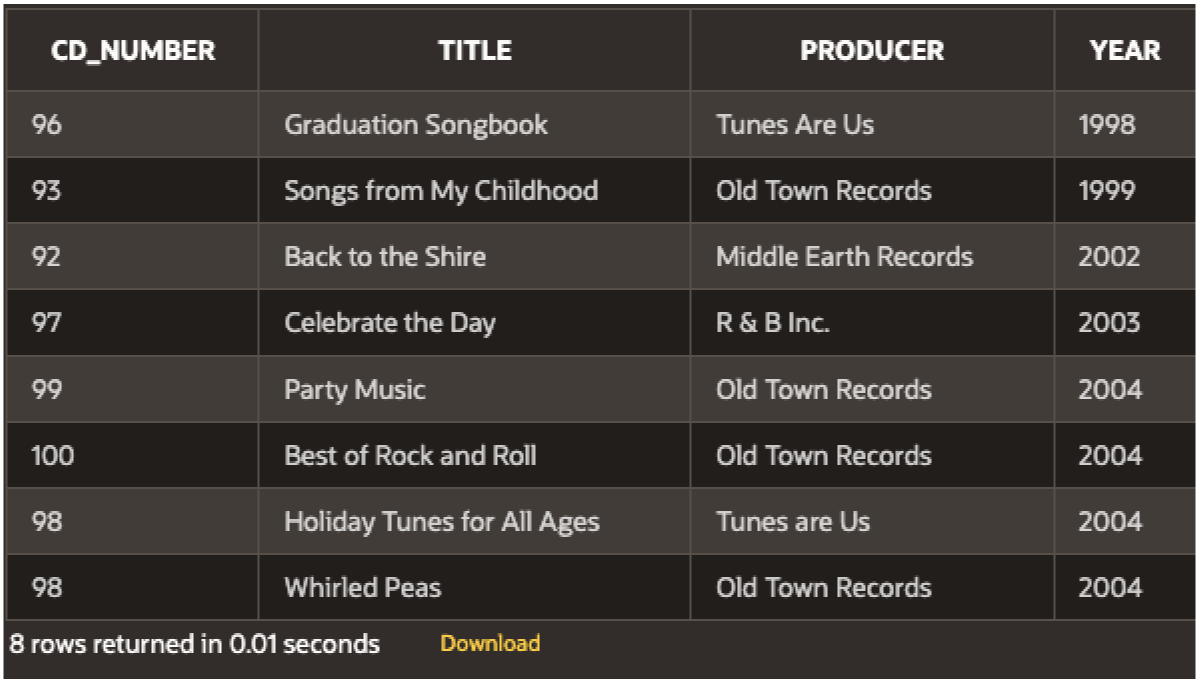
```
DELETE FROM read_copy_d_cds
WHERE year = '2001';
```

10. Execute a `SELECT *` statement for the base table `copy_d_cds`. What rows were deleted?

ANS:

```
SELECT * FROM copy_d_cds
ORDER BY year;
```

The rows deleted were the rows that had cd_number 90 and year 2000 and 2001.



CD_NUMBER	TITLE	PRODUCER	YEAR
96	Graduation Songbook	Tunes Are Us	1998
93	Songs from My Childhood	Old Town Records	1999
92	Back to the Shire	Middle Earth Records	2002
97	Celebrate the Day	R & B Inc.	2003
99	Party Music	Old Town Records	2004
100	Best of Rock and Roll	Old Town Records	2004
98	Holiday Tunes for All Ages	Tunes are Us	2004
98	Whirled Peas	Old Town Records	2004

8 rows returned in 0.01 seconds

11. What are the restrictions on modifying data through a view?
    a. **ANS:**
        i. Data can be modified directly through a simple view, but not always for a complex view
        ii. A row cannot be removed from an underlying base table if the view contains any of the following:
            1. Group functions
            2. GROUP BY clause
            3. DISTINCT keyword
            4. The pseudocolumn ROWNUM keyword

        iii. Alongside the restriction in bullet ii, data cannot be modified through a view if the view contains…
            1. Columns defined by expressions
        iv. Alongside the restriction in bullet ii and iii, view cannot have data ADDED if it…
            1. Does not include NOT NULL columns in the base table

12. What is Moore's Law? Do you consider that it will continue to apply indefinitely? Support your opinion with research from the internet.
    a. Moore's Law is the observation that the amount of components in a single chip will double every two years with little cost increase. Increasing the amount of transistors in a circuit means that the transistor is becoming smaller and smaller to fit them all in. Thus, I believe that Moore's Law will not apply indefinitely as it will likely become harder to shrink transistors at a certain point such as making them smaller than an atom.
        i. Source: https://www.investopedia.com/terms/m/mooreslaw.asp

13. What is the "singularity" in terms of computing?
    a. "Singularity" in terms of computing refers to a theoretical scenario where growth in technology becomes uncontrollable and irreversible. For example, technology can no longer be understood through existing models and machines may surpass human intelligence to the point where they are autonomous and innovate far beyond what humans were capable of.
        i. Source: https://www.ibm.com/think/topics/technological-singularity

15-3 Managing Views

Word	Definition
TOP-N ANALYSIS	Asks for the N largest or smallest values in a column
DROP VIEW	Removes a view
INLINE VIEW	Subquery with an alias that can be used within a SQL statement

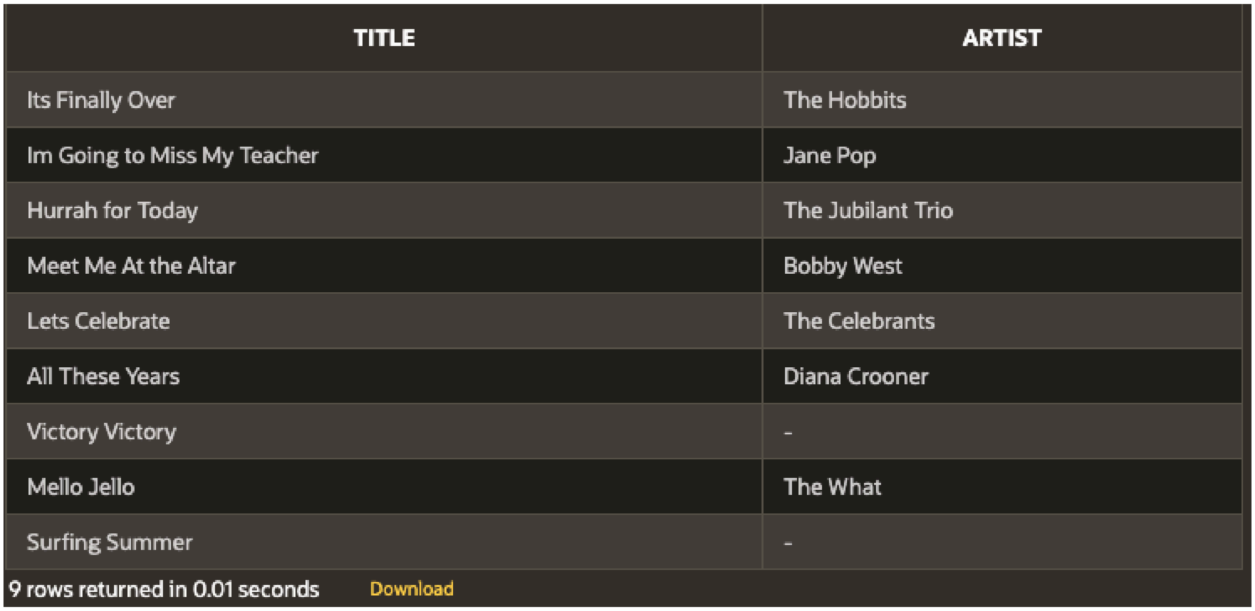
1. Create a view from the `copy_d_songs` table called `view_copy_d_songs` that includes only the title and artist. Execute a `SELECT *` statement to verify that the view exists.

ANS:

CREATE OR REPLACE VIEW view_copy_d_songs

```
AS SELECT title, artist
    FROM copy_d_songs
```

```
SELECT * FROM view_copy_d_songs;
```



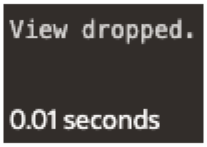
TITLE	ARTIST
Its Finally Over	The Hobbits
Im Going to Miss My Teacher	Jane Pop
Hurrah for Today	The Jubilant Trio
Meet Me At the Altar	Bobby West
Lets Celebrate	The Celebrants
All These Years	Diana Crooner
Victory Victory	-
Mello Jello	The What
Surfing Summer	-

9 rows returned in 0.01 seconds

2. Issue a `DROP view_copy_d_songs`. Execute a `SELECT *` statement to verify that the view has been deleted.

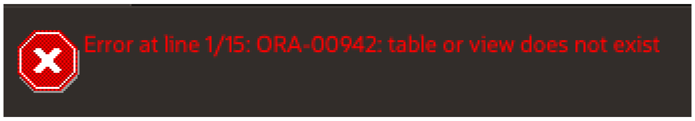
ANS:

```
DROP VIEW view_copy_d_songs;
```



```
View dropped.

0.01 seconds
```

```
SELECT * FROM view_copy_d_songs;
```



```
Error at line 1/15: ORA-00942: table or view does not exist
```

3. Create a query that selects the last name and salary from the Oracle database. Rank the salaries from highest to lowest for the top three employees.

ANS:

Version 1

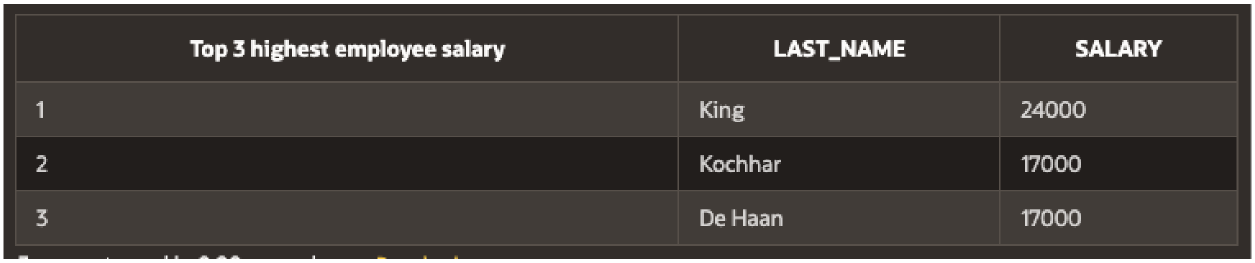
```
SELECT ROWNUM AS "Top 3 highest employee salary", last_name, salary
```

```
FROM employees
WHERE ROWNUM <= 3
ORDER BY salary DESC, "Top 3 highest employee salary"
```

NOTE: "Top 3 highest employee salary" was added in the ORDER BY statement because two employees have the same salary and without it, the order of it would be 1, 3, 2 rather than 1, 2, 3

Version 2

```
SELECT ROWNUM AS "Top 3 highest employee salary", last_name, salary
FROM (
   SELECT last_name, salary
   FROM employees
   ORDER BY salary DESC
)
WHERE ROWNUM <= 3
```



Top 3 highest employee salary	LAST_NAME	SALARY
1	King	24000
2	Kochhar	17000
3	De Haan	17000

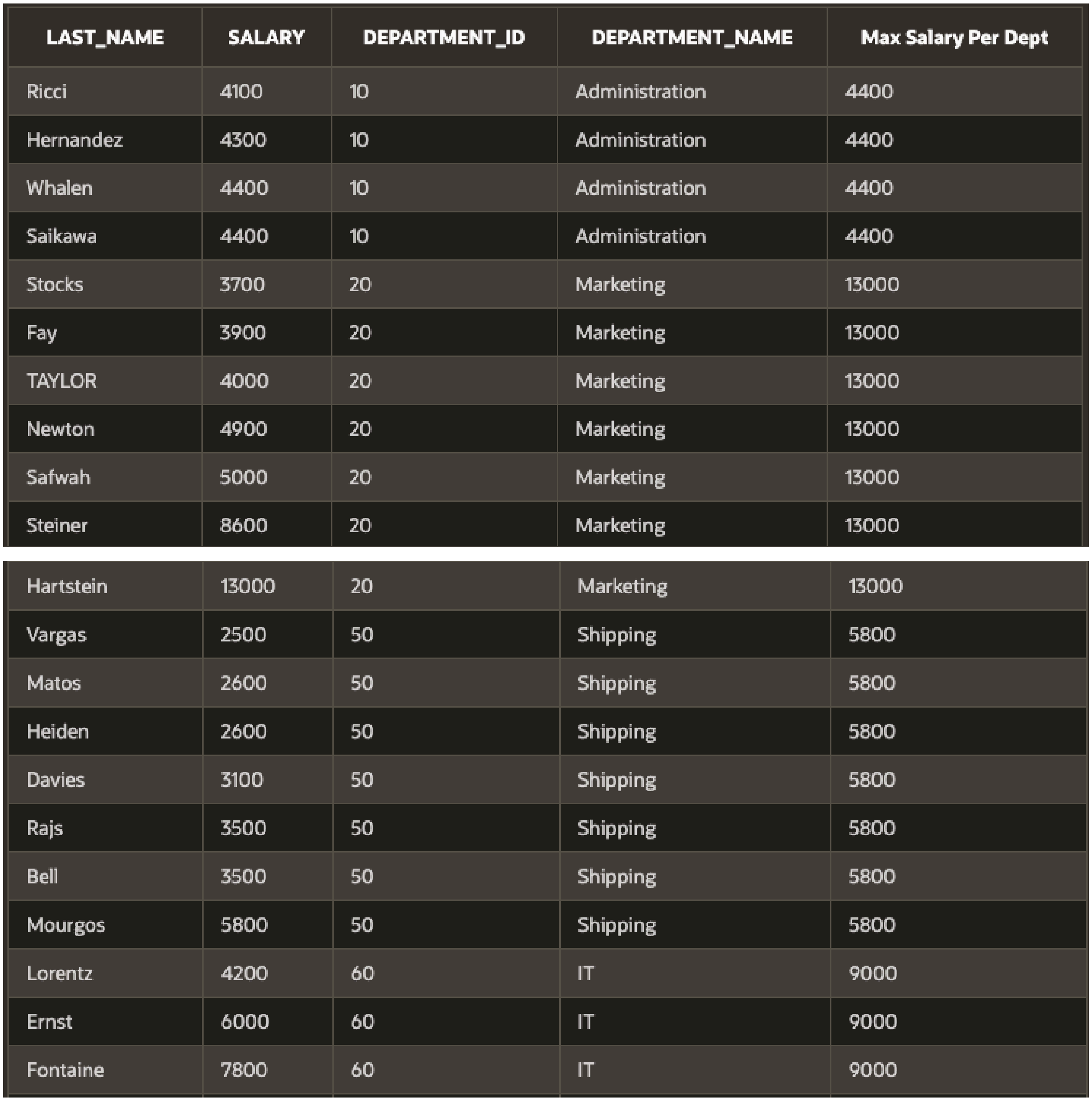
4. Construct an inline view from the Oracle database that lists the last name, salary, department ID, and maximum salary for each department. Hint: One query will need to calculate maximum salary by department ID.

ANS:

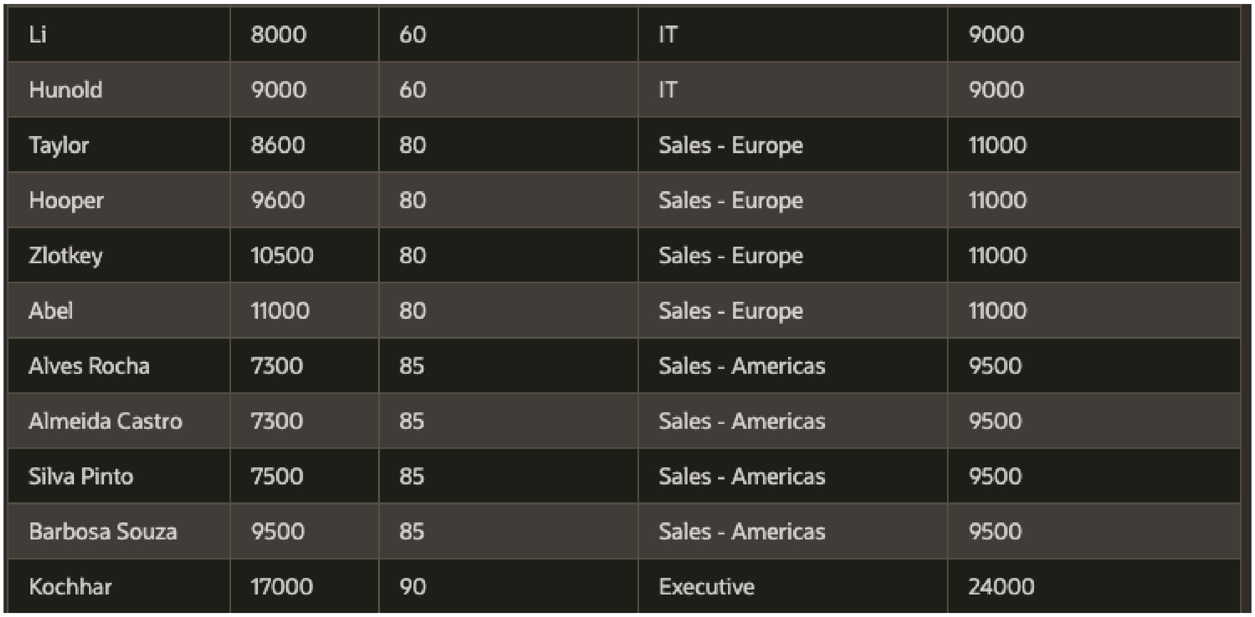
NOTE: I added department_name because I feel having it would make it more clear as to which exact department has that max salary.

```
SELECT e.last_name, e.salary, e.department_id, dn.department_name, d.max_sal "Max Salary Per Dept"
FROM employees e,
   (SELECT department_id, MAX(salary) max_sal
    FROM employees e
    GROUP BY department_id) d,
   (SELECT department_id, department_name
    FROM departments) dn
WHERE e.department_id = d.department_id
```

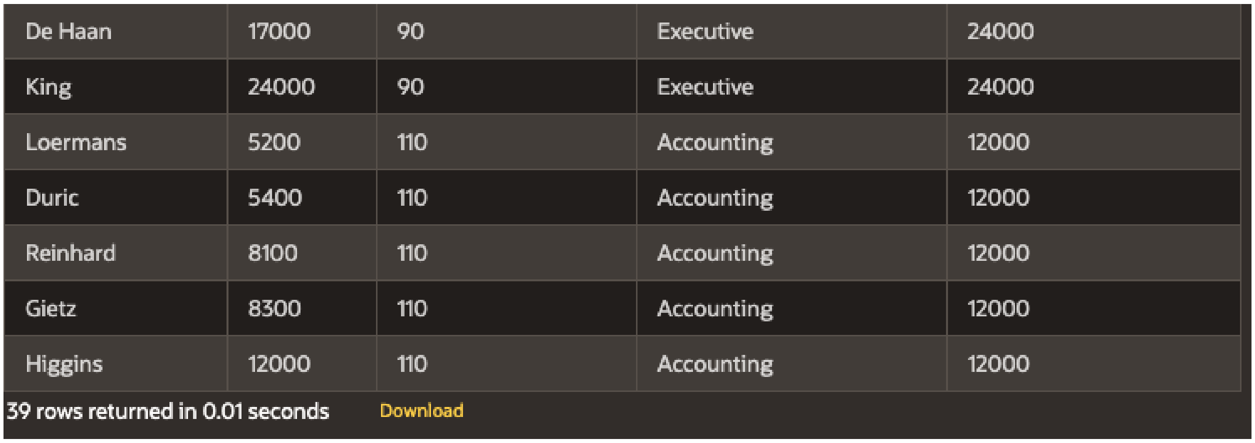
```
    AND e.department_id = dn.department_id
ORDER BY e.department_id, e.salary;
```



LAST_NAME	SALARY	DEPARTMENT_ID	DEPARTMENT_NAME	Max Salary Per Dept
Ricci	4100	10	Administration	4400
Hernandez	4300	10	Administration	4400
Whalen	4400	10	Administration	4400
Saikawa	4400	10	Administration	4400
Stocks	3700	20	Marketing	13000
Fay	3900	20	Marketing	13000
TAYLOR	4000	20	Marketing	13000
Newton	4900	20	Marketing	13000
Safwah	5000	20	Marketing	13000
Steiner	8600	20	Marketing	13000
Hartstein	13000	20	Marketing	13000
Vargas	2500	50	Shipping	5800
Matos	2600	50	Shipping	5800
Heiden	2600	50	Shipping	5800
Davies	3100	50	Shipping	5800
Rajs	3500	50	Shipping	5800
Bell	3500	50	Shipping	5800
Mourgos	5800	50	Shipping	5800
Lorentz	4200	60	IT	9000
Ernst	6000	60	IT	9000
Fontaine	7800	60	IT	9000



Li	8000	60	IT	9000
Hunold	9000	60	IT	9000
Taylor	8600	80	Sales - Europe	11000
Hooper	9600	80	Sales - Europe	11000
Zlotkey	10500	80	Sales - Europe	11000
Abel	11000	80	Sales - Europe	11000
Alves Rocha	7300	85	Sales - Americas	9500
Almeida Castro	7300	85	Sales - Americas	9500
Silva Pinto	7500	85	Sales - Americas	9500
Barbosa Souza	9500	85	Sales - Americas	9500
Kochhar	17000	90	Executive	24000



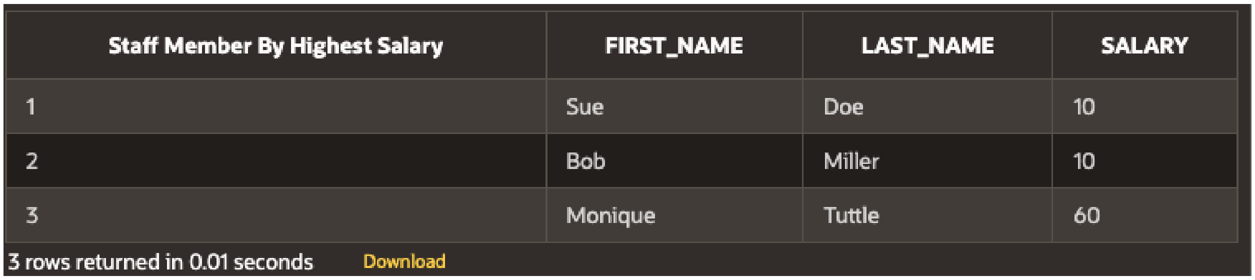
De Haan	17000	90	Executive	24000
King	24000	90	Executive	24000
Loermans	5200	110	Accounting	12000
Duric	5400	110	Accounting	12000
Reinhard	8100	110	Accounting	12000
Gietz	8300	110	Accounting	12000
Higgins	12000	110	Accounting	12000

39 rows returned in 0.01 seconds

5. Create a query that will return the staff members of Global Fast Foods ranked by salary from lowest to highest.

ANS:

```
SELECT ROWNUM AS "Staff Member By Highest Salary", first_name, last_name, salary
FROM (
   SELECT first_name, last_name, salary
   FROM copy_f_staffs
   ORDER BY salary
)
```



Staff Member By Highest Salary	FIRST_NAME	LAST_NAME	SALARY
1	Sue	Doe	10
2	Bob	Miller	10
3	Monique	Tuttle	60

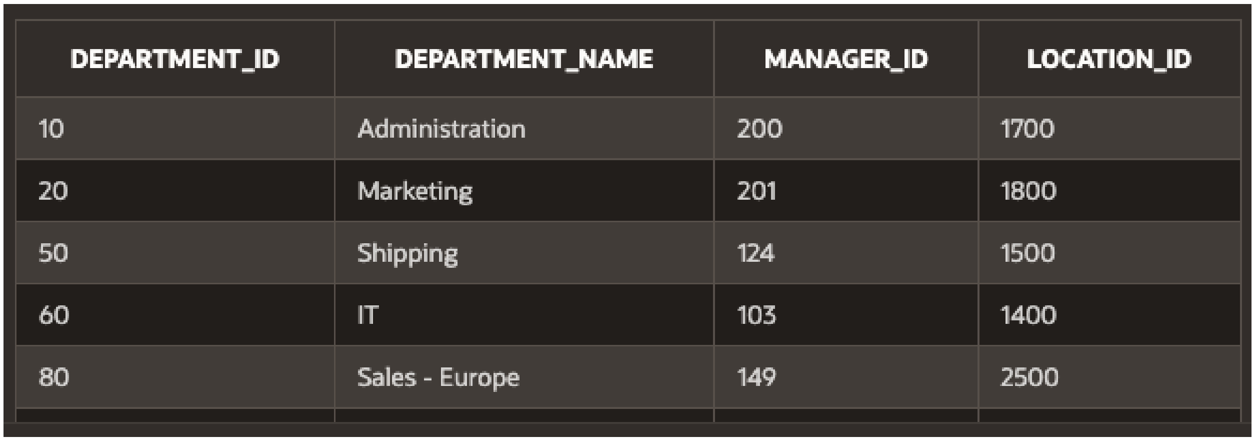
3 rows returned in 0.01 seconds Download

Extension Exercises

1. Create a new table called my_departments and add all columns and all rows to it using a subquery from the Oracle departments table. Do a SELECT * from my_departments to confirm that you have all the columns and rows.

ANS:
CREATE TABLE my_departments
AS (SELECT * FROM departments);

SELECT * FROM my_departments;



DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales - Europe	149	2500

2. To view any constraints that may affect the my_departments table, DESCRIBE my_departments to check if any constraints were carried over from the departments table. If there are constraints on my_departments, use an ALTER TABLE command to DISABLE all constraints on my_departments.

ANS:
DESCRIBE my_departments;

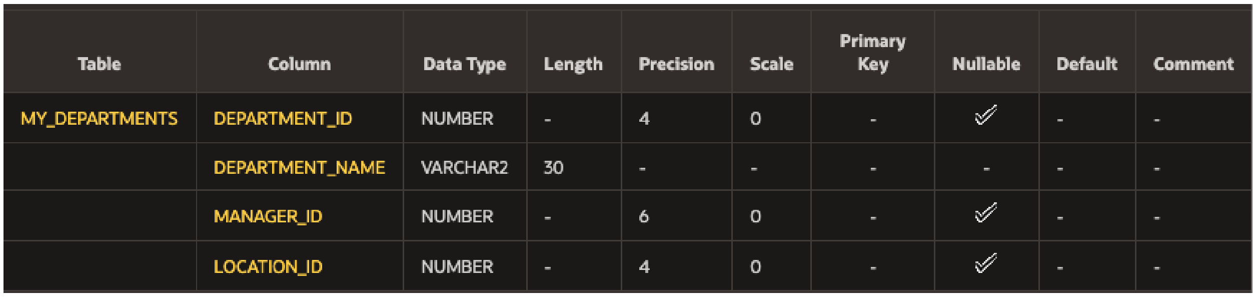
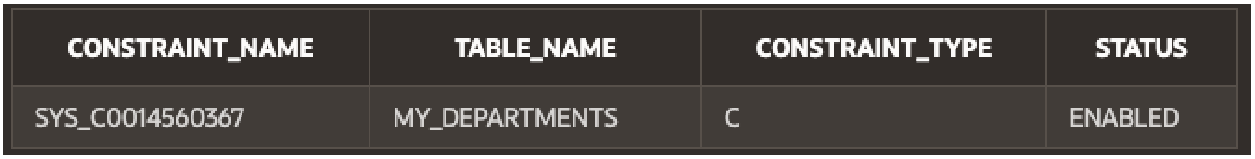


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MY_DEPARTMENTS	DEPARTMENT_ID	NUMBER	-	4	0	-	✓	-	-
	DEPARTMENT_NAME	VARCHAR2	30	-	-	-	-	-	-
	MANAGER_ID	NUMBER	-	6	0	-	✓	-	-
	LOCATION_ID	NUMBER	-	4	0	-	✓	-	-

There are NOT NULL constraints present in my_departments table.

<u>To view the constraint name</u>

SELECT constraint_name, table_name, constraint_type, status
FROM USER_CONSTRAINTS
WHERE LOWER(table_name) ='my_departments';



CONSTRAINT_NAME	TABLE_NAME	CONSTRAINT_TYPE	STATUS
SYS_C0014560367	MY_DEPARTMENTS	C	ENABLED

<u>Disable constraint</u>

ALTER TABLE my_departments
DISABLE CONSTRAINT SYS_C0014560367;

<u>NOT NULL constraint is now disabled</u>

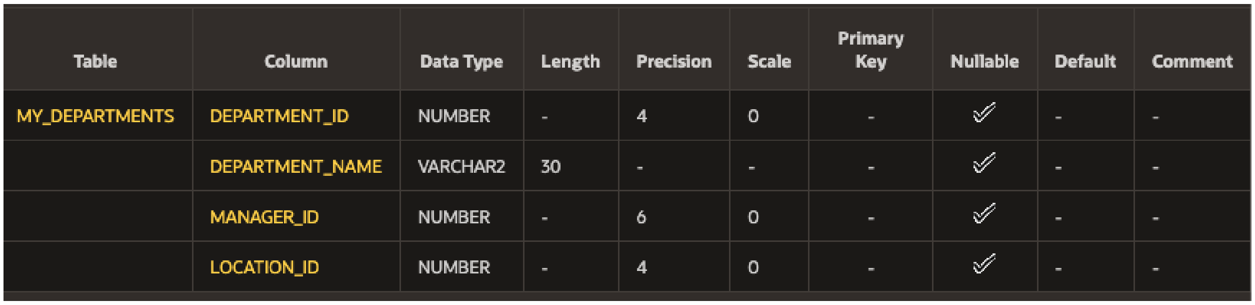


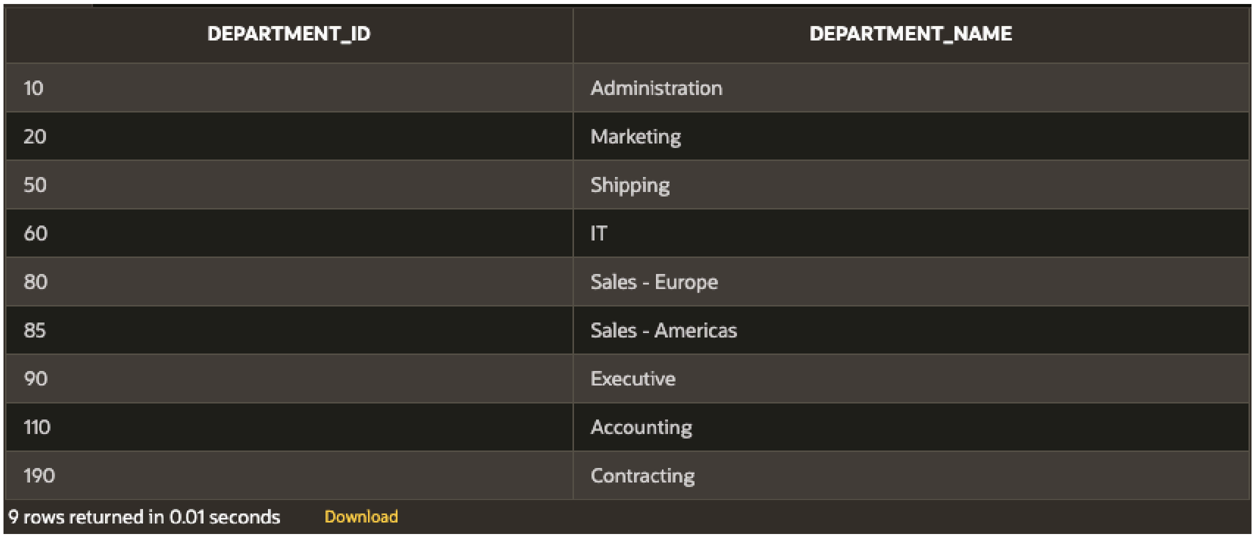
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MY_DEPARTMENTS	DEPARTMENT_ID	NUMBER	-	4	0	-	✓	-	-
	DEPARTMENT_NAME	VARCHAR2	30	-	-	-	✓	-	-
	MANAGER_ID	NUMBER	-	6	0	-	✓	-	-
	LOCATION_ID	NUMBER	-	4	0	-	✓	-	-

3. Create a view called view_my_departments that includes department_id and department_name.

ANS:

CREATE VIEW view_my_departments
AS SELECT department_id, department_name
   FROM my_departments;

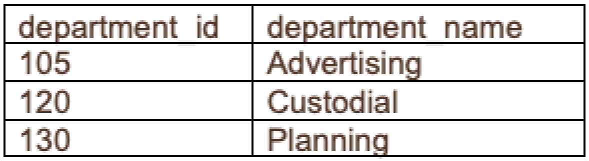
SELECT * FROM view_my_departments;



DEPARTMENT_ID	DEPARTMENT_NAME
10	Administration
20	Marketing
50	Shipping
60	IT
80	Sales - Europe
85	Sales - Americas
90	Executive
110	Accounting
190	Contracting

9 rows returned in 0.01 seconds

4. Add the following data to the my_departments table using view_my_departments.



department_id	department_name
105	Advertising
120	Custodial
130	Planning

ANS:

```
INSERT INTO view_my_departments
   (department_id, department_name)
VALUES
   X
```

List of X used

- (105, 'Advertising');
- (120, 'Custodial');
- (130, 'Planning');

5. Create or enable the `department_id` column as the primary key.

ANS:

```
ALTER TABLE my_departments
ADD CONSTRAINT my_depts_dept_id_pk PRIMARY KEY (department_id);

DESCRIBE my_departments;
```

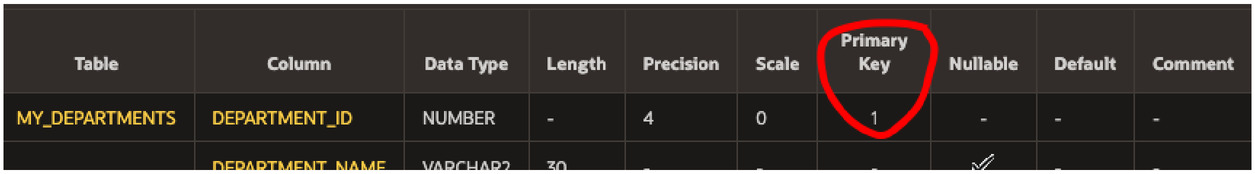
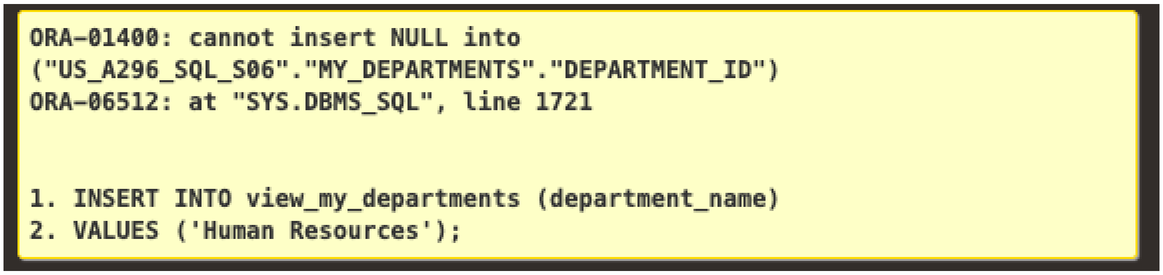


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MY_DEPARTMENTS	DEPARTMENT_ID	NUMBER	-	4	0	1	-	-	-
	DEPARTMENT_NAME	VARCHAR2	30	-	-	-	✓	-	-

6. Enter a new department named **Human Resources** into the `my_departments` table using `view_my_departments`. Do not add a new department ID.

ANS:
INSERT INTO view_my_departments (department_name)
VALUES ('**Human Resources**');

An error results as expected since primary keys cannot be NULL



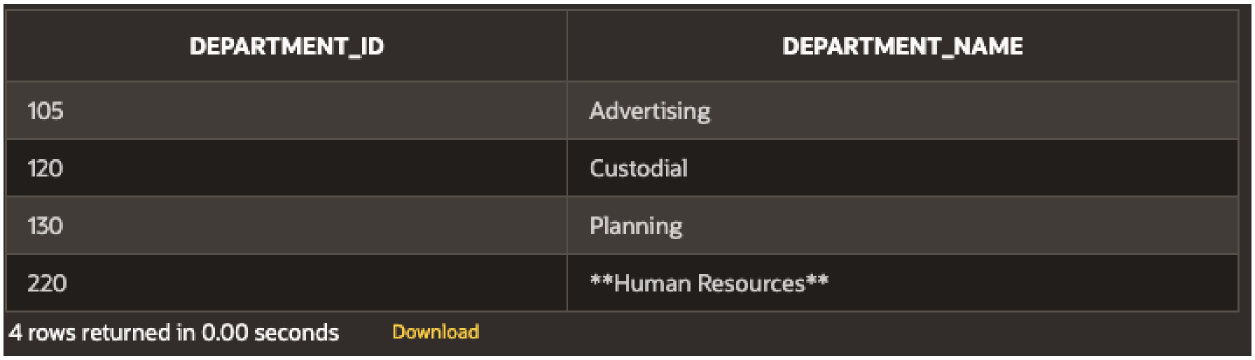
```
ORA-01400: cannot insert NULL into
("US_A296_SQL_S06"."MY_DEPARTMENTS"."DEPARTMENT_ID")
ORA-06512: at "SYS.DBMS_SQL", line 1721


1. INSERT INTO view_my_departments (department_name)
2. VALUES ('Human Resources');
```

7. Add the **Human Resources** department, `department_id` 220, to `my_departments` using `view_my_departments`.

ANS:
INSERT INTO view_my_departments
   (department_id, department_name)
VALUES
   (220, '**Human Resources**');

8. Verify that the new additions to `my_departments` were added using `view_my_departments`.

ANS:
SELECT *
FROM view_my_departments
WHERE department_id IN (105, 120, 130, 220)
ORDER BY department_id;



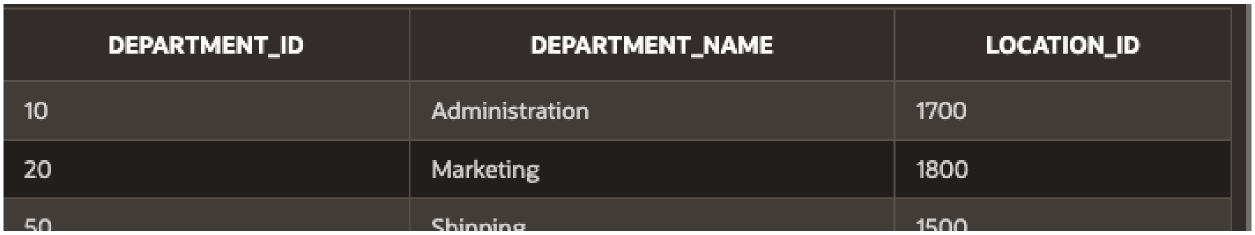
DEPARTMENT_ID	DEPARTMENT_NAME
105	Advertising
120	Custodial
130	Planning
220	**Human Resources**

4 rows returned in 0.00 seconds

9. Modify `view_my_departments` to include `location_id`. Execute a `SELECT *` command to show what columns are present and a `DESCRIBE` command to view the columns and associated constraints.

ANS:

```
CREATE OR REPLACE VIEW view_my_departments
AS SELECT department_id, department_name, location_id
   FROM my_departments;
```



DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
10	Administration	1700
20	Marketing	1800
50	Shipping	1500

DESCRIBE view_my_departments;

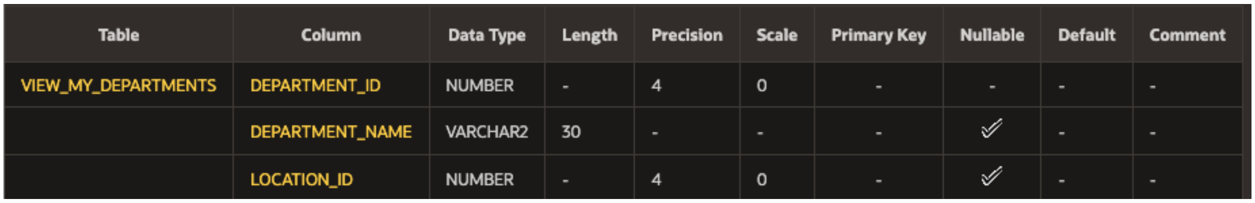


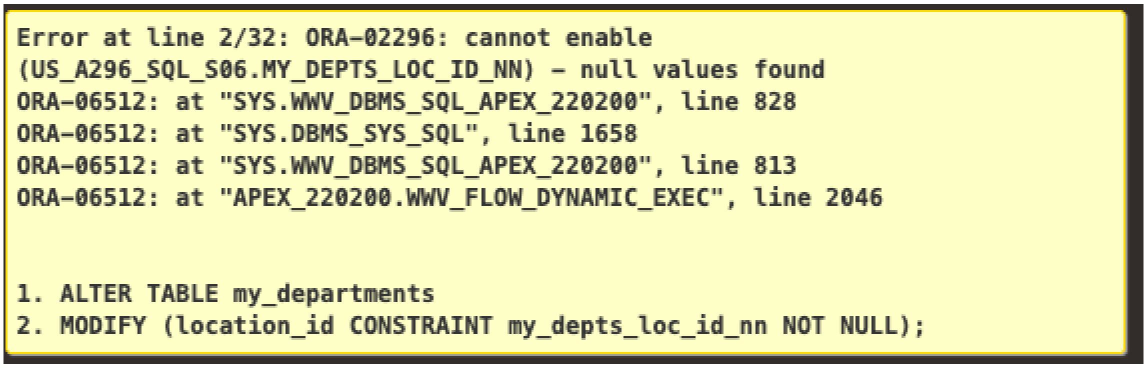
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
VIEW_MY_DEPARTMENTS	DEPARTMENT_ID	NUMBER	-	4	0	-	-	-	-
	DEPARTMENT_NAME	VARCHAR2	30	-	-	-	✓	-	-
	LOCATION_ID	NUMBER	-	4	0	-	✓	-	-

10. Make `location_id` a `NOT NULL` column in the `my_departments` table.

ANS:

```
ALTER TABLE my_departments
MODIFY (location_id CONSTRAINT my_depts_loc_id_nn NOT NULL);
```

Since the location_id already has NULL values, this error pops up



```
Error at line 2/32: ORA-02296: cannot enable
(US_A296_SQL_S06.MY_DEPTS_LOC_ID_NN) - null values found
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_220200", line 828
ORA-06512: at "SYS.DBMS_SYS_SQL", line 1658
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_220200", line 813
ORA-06512: at "APEX_220200.WWV_FLOW_DYNAMIC_EXEC", line 2046


1. ALTER TABLE my_departments
2. MODIFY (location_id CONSTRAINT my_depts_loc_id_nn NOT NULL);
```

To fix this, all NULL values in location_id should be changed

```
UPDATE my_departments
SET location_id = 0
WHERE location_id IS NULL;
```

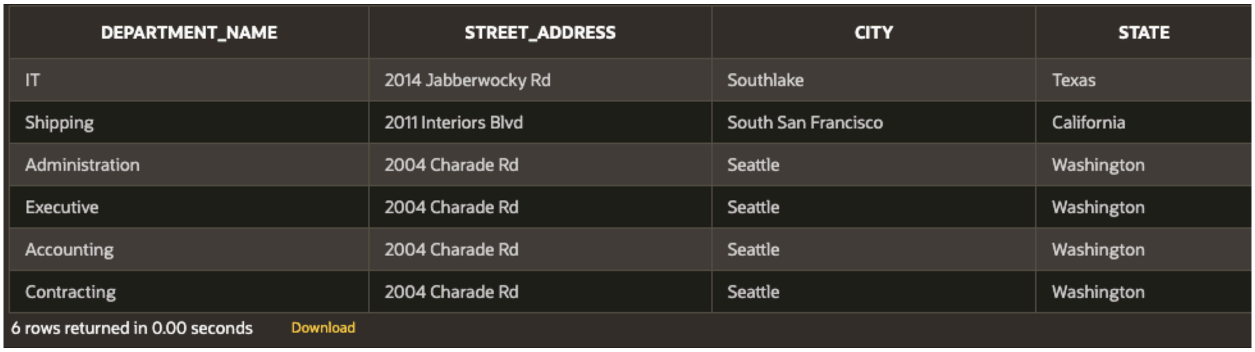
Adding the NOT NULL constraint to the location_id column will now work

```
ALTER TABLE my_departments
MODIFY (location_id CONSTRAINT my_depts_loc_id_nn NOT NULL);
```

11. Using the Oracle database, create a complex view between `locations` and `departments` with only the following columns: `department_name`, `street_address`, `city`, and `state`. Include only U.S. cities. Verify that the view was created using a `SELECT *` statement.

ANS:

```
CREATE OR REPLACE VIEW view_depts_locs
AS (
   SELECT d.department_name, l.street_address, l.city, l.state_province "state"
   FROM departments_pl d INNER JOIN locations l
   ON d.location_id = l.location_id
      AND l.country_id IN (
         SELECT country_id
         FROM countries
         WHERE LOWER(country_name) LIKE 'united state'
      )
)

SELECT * FROM view_depts_locs
```



DEPARTMENT_NAME	STREET_ADDRESS	CITY	STATE
IT	2014 Jabberwocky Rd	Southlake	Texas
Shipping	2011 Interiors Blvd	South San Francisco	California
Administration	2004 Charade Rd	Seattle	Washington
Executive	2004 Charade Rd	Seattle	Washington
Accounting	2004 Charade Rd	Seattle	Washington
Contracting	2004 Charade Rd	Seattle	Washington

6 rows returned in 0.00 seconds Download

16-1 Working With Sequences

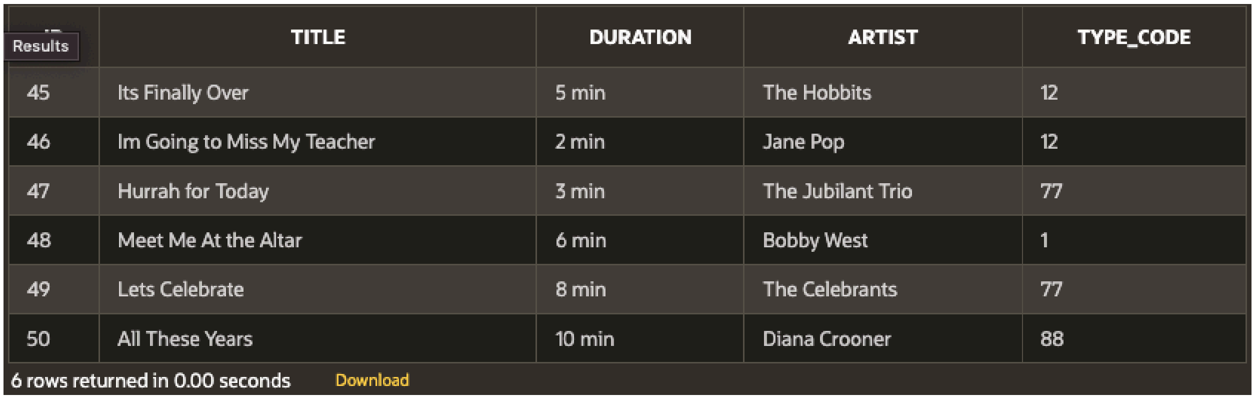
Word	Definition
CREATE SEQUENCE	Command that automatically generates sequential numbers
Sequences	Generates a numeric value
NEXTVAL	Returns the next available sequence value
INCREMENT BY	Specifies the interval between sequence numbers
NOMAXVALUE	Specifies a maximum value of 10^27 for an ascending sequence and -1 for a descending sequence (default)
CURRVAL	returns the current sequence value
MINVALUE n	specifies the minimum sequence value
CYCLE | NOCYCLE	specifies whether the sequence continues to generate values after reaching its maximum or minimum values
NOMINVALUE	specifies a minimum value of 1 for an ascending sequence and -(10^26) for a descending sequence (default)
MAXVALUE n	specifies a maximum or default value the sequence can generate
START WITH	specifies the first sequence number to be generated
CACHE n | NOCACHE	specifies how many values the Server pre-allocates and keeps in memory

1. Using CREATE TABLE AS subquery syntax, create a seq_d_songs table of all the columns in the DJs on Demand database table d_songs. Use the SELECT * in the subquery to make sure that you have copied all of the columns.

ANS:
CREATE TABLE seq_d_songs

AS (SELECT * FROM d_songs);

SELECT * FROM seq_d_songs;



Results	TITLE	DURATION	ARTIST	TYPE_CODE
45	Its Finally Over	5 min	The Hobbits	12
46	Im Going to Miss My Teacher	2 min	Jane Pop	12
47	Hurrah for Today	3 min	The Jubilant Trio	77
48	Meet Me At the Altar	6 min	Bobby West	1
49	Lets Celebrate	8 min	The Celebrants	77
50	All These Years	10 min	Diana Crooner	88

6 rows returned in 0.00 seconds

2. Because you are using copies of the original tables, the only constraints that were carried over were the NOT NULL constraints. Create a sequence to be used with the primary-key column of the seq_d_songs table. To avoid assigning primary-key numbers to these tables that already exist, the sequence should start at 100 and have a maximum value of 1000. Have your sequence increment by 2 and have NOCACHE and NOCYCLE. Name the sequence seq_d_songs_seq.

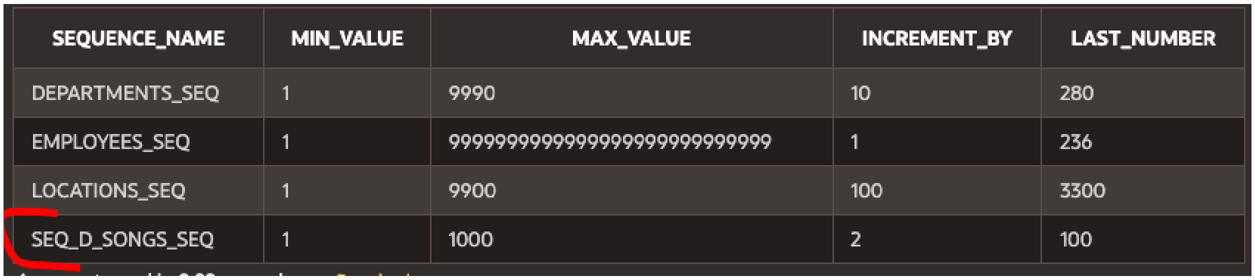
ANS:

```
CREATE SEQUENCE seq_d_songs_seq
   INCREMENT BY 2
   START WITH 100
   MAXVALUE 1000
   NOCACHE
   NOCYCLE;
```

3. Query the USER_SEQUENCES data dictionary to verify the seq_d_songs_seq SEQUENCE settings.

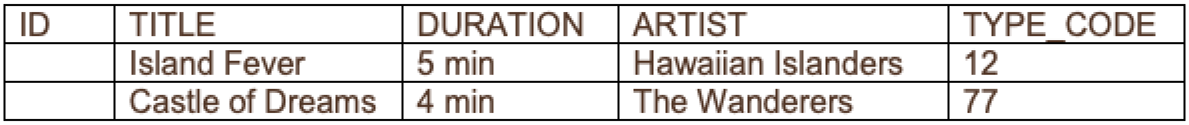
ANS:

```
SELECT sequence_name, min_value, max_value, increment_by, last_number
FROM USER_SEQUENCES;
```



SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
DEPARTMENTS_SEQ	1	9990	10	280
EMPLOYEES_SEQ	1	9999999999999999999999999999	1	236
LOCATIONS_SEQ	1	9900	100	3300
SEQ_D_SONGS_SEQ	1	1000	2	100

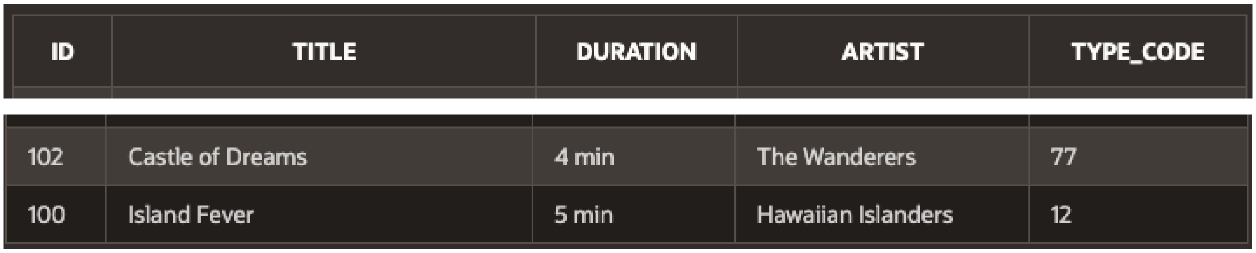
4. Insert two rows into the seq_d_songs table. Be sure to use the sequence that you created for the ID column. Add the two songs shown in the graphic.



ID	TITLE	DURATION	ARTIST	TYPE_CODE
	Island Fever	5 min	Hawaiian Islanders	12
	Castle of Dreams	4 min	The Wanderers	77

ANS:

```
INSERT INTO seq_d_songs
   (id, title, duration, artist, type_code)
VALUES
   (seq_d_songs_seq.NEXTVAL, 'Island Fever', '5 min', 'Hawaiian Islanders', 12)

INSERT INTO seq_d_songs
   (id, title, duration, artist, type_code)
VALUES
   (seq_d_songs_seq.NEXTVAL, 'Castle of Dreams', '4 min', 'The Wanderers', 77)
```



ID	TITLE	DURATION	ARTIST	TYPE_CODE
102	Castle of Dreams	4 min	The Wanderers	77
100	Island Fever	5 min	Hawaiian Islanders	12

5. Write out the syntax for seq_d_songs_seq to view the current value for the sequence. Use the DUAL table. (Oracle Application Developer will not run this query.)

ANS:

```
SELECT seq_d_songs_seq.CURRVAL
FROM DUAL;
```

6. What are three benefits of using SEQUENCEs?

a. Automatically creates unique numbers that eliminates the issue of duplicate numbers for fields that require non-duplicates, such as for a primary key

b. Reduces the amount of code that needs to be written

c. Is shareable, so multiple users can access it

7. What are the advantages of caching sequence values?
   a. Faster access to sequence values
8. Name three reasons why gaps may occur in a sequence?
   a. Rolling back a statement that has a sequence causes that number to be lost
   b. A system crash causes numbers in a sequence to be lost if those values are cached into memory
   c. Using the same sequence in multiple tables can cause gaps

Extension Exercise

1. Create a table called “students”. You can decide which columns belong in that table and what datatypes these columns require. (The students may create a table with different columns; however, the important piece that must be there is the student_id column with a numeric datatype. This column length must allow the sequence to fit, e.g. a column length of 4 with a sequence that starts with 1 and goes to 10000000 will not work after student #9999 is entered.)

ANS:

```
CREATE TABLE students_other (
   student_id NUMBER CONSTRAINT stu_others_stu_id_pk PRIMARY KEY,
   first_name VARCHAR2(30),
   last_name VARCHAR2(30),
   grade NUMBER
)
```

2. Create a sequence called student_id_seq so that you can assign unique student_id numbers for all students that you add to your table.

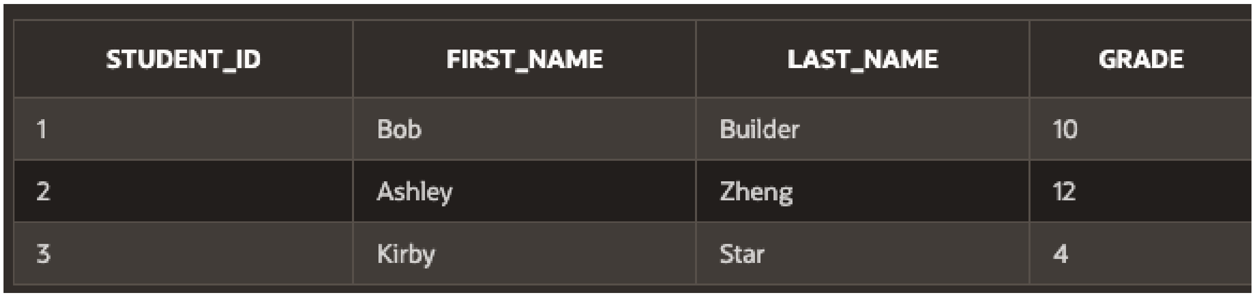
ANS:

```
CREATE SEQUENCE students_other_seq
   INCREMENT BY 1
   START WITH 1
   MAXVALUE 10000000
   NOCACHE
   NOCYCLE
```

3. Now write the code to add students to your STUDENTS table, using your sequence “database object.

ANS:

```
INSERT INTO students_other
   (student_id, first_name, last_name, grade)
VALUES
   (students_other_seq.NEXTVAL, 'some_first_name', 'some_last_name', some_number);

SELECT * FROM students_other;
```



STUDENT_ID	FIRST_NAME	LAST_NAME	GRADE
1	Bob	Builder	10
2	Ashley	Zheng	12
3	Kirby	Star	4

16-2 Indexes and Synonyms

Word	Definition
Confirming index	Confirms the existence of indexes from the USER_INDEXES data dictionary view
Index	Schema object that speeds up retrieval of rows
CREATE PUBLIC SYNONYM	To refer to a table by another name to simplify access
Composite index	An index that you create on multiple columns in a table
Unique index	The Oracle Server automatically creates this index when you define a column in a table to have a PRIMARY KEY or a UNIQUE KEY constraint
Function-based index	Stores the indexed values and uses the index based on a SELECT statement to retrieve the data
DROP INDEX	Removes an index
Synonym	Gives alternative names to objects

1. What is an index and what is it used for?

    a. An index is a schema object that speeds up row retrieval from a table by using a pointer. Without it, a full table scan occurs to get the desired info which can be slower.

2. What is a ROWID, and how is it used?
    a. ROWID is a base 64 string representation of the row address, which contains a block identifier, row location in that block, and the database file identifier. Indexes use ROWID because they are the fastest way to access any particular row.
3. When will an index be created automatically?
    a. An index is made automatically when a column is given a PRIMARY KEY or UNIQUE KEY constraint
4. Create a nonunique index (foreign key) for the DJs on Demand column (cd_number) in the D_TRACK_LISTINGS table. Use the Oracle Application Developer SQL Workshop Data Browser to confirm that the index was created.

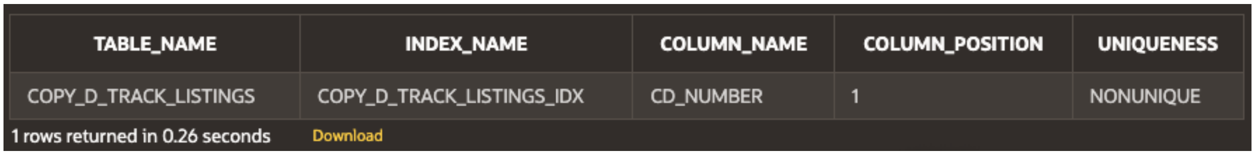
ANS:

```
CREATE INDEX copy_d_track_listings_idx
ON copy_d_track_listings(cd_number);
```

Verify Index Through Query

```
SELECT DISTINCT id.table_name, ic.index_name, ic.column_name, ic.column_position, id.uniqueness
FROM USER_INDEXES id, USER_IND_COLUMNS ic
WHERE id.table_name = ic.table_name
AND LOWER(ic.table_name) = 'copy_d_track_listings';
```

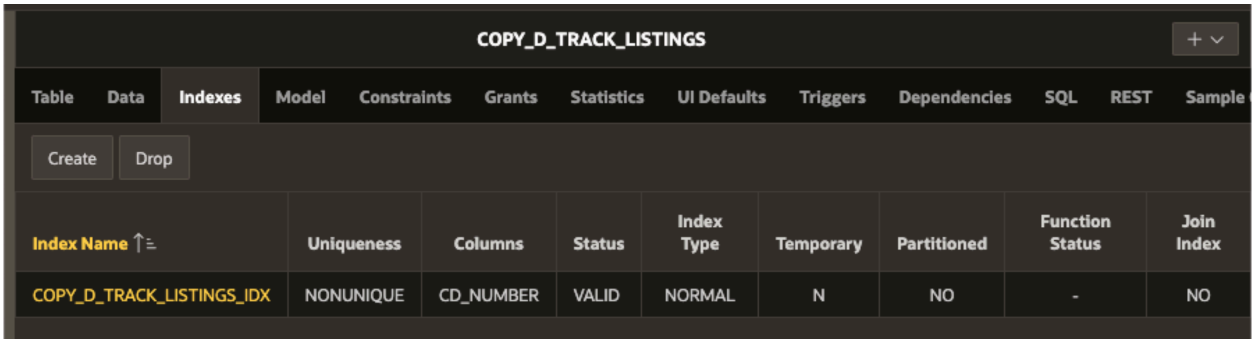
NOTE: Using LOWER() is crucial or else no data will appear. The reason is b/c the table names are all uppercase in the USER… tables. Alternatively, ic.table_name = ‘COPY_D_TRACK_LISTINGS’ would work too.



TABLE_NAME	INDEX_NAME	COLUMN_NAME	COLUMN_POSITION	UNIQUENESS
COPY_D_TRACK_LISTINGS	COPY_D_TRACK_LISTINGS_IDX	CD_NUMBER	1	NONUNIQUE

1 rows returned in 0.26 seconds

Verify Index Through Oracle Application Developer SQL Workshop Data/Object Browser



Index Name ↑	Uniqueness	Columns	Status	Index Type	Temporary	Partitioned	Function Status	Join Index
COPY_D_TRACK_LISTINGS_IDX	NONUNIQUE	CD_NUMBER	VALID	NORMAL	N	NO	-	NO

5. Use the join statement to display the indexes and uniqueness that exist in the data dictionary for the DJs on Demand D_SONGS table.

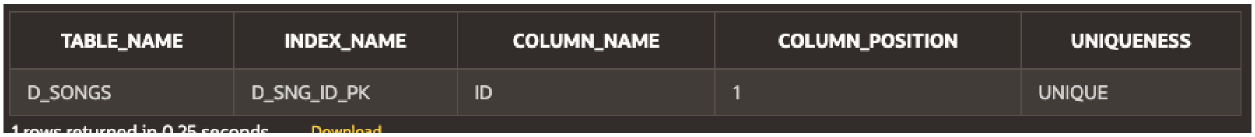
ANS:

Version 1

SELECT DISTINCT id.table_name, ic.index_name, ic.column_name, ic.column_position, id.uniqueness
FROM USER_INDEXES id, USER_IND_COLUMNS ic
WHERE id.table_name = ic.table_name
AND LOWER(ic.table_name) = 'd_songs';

Version 2

SELECT DISTINCT id.table_name, ic.index_name, ic.column_name, ic.column_position, id.uniqueness
FROM USER_INDEXES id INNER JOIN USER_IND_COLUMNS ic
ON id.table_name = ic.table_name
  AND LOWER(ic.table_name) = 'd_songs';



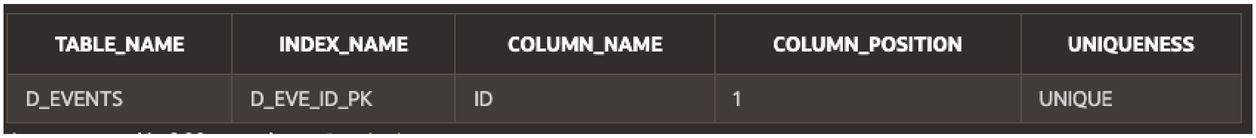
TABLE_NAME	INDEX_NAME	COLUMN_NAME	COLUMN_POSITION	UNIQUENESS
D_SONGS	D_SNG_ID_PK	ID	1	UNIQUE

1 rows returned in 0.25 seconds   Download

6. Use a SELECT statement to display the index_name, table_name, and uniqueness from the data dictionary USER_INDEXES for the DJs on Demand D_EVENTS table.

ANS:

SELECT DISTINCT table_name, index_name, uniqueness
FROM USER_INDEXES
WHERE LOWER(table_name) = 'd_events';

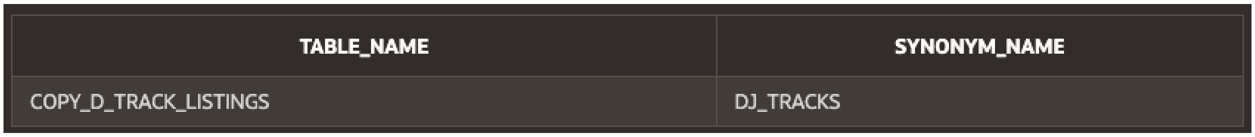


TABLE_NAME	INDEX_NAME	COLUMN_NAME	COLUMN_POSITION	UNIQUENESS
D_EVENTS	D_EVE_ID_PK	ID	1	UNIQUE

7. Write a query to create a synonym called dj_tracks for the DJs on Demand d_track_listings table.

ANS:

```
CREATE SYNONYM dj_tracks
FOR copy_d_track_listings;

SELECT table_name, synonym_name
FROM USER_SYNONYMS
WHERE LOWER(table_name) = 'copy_d_track_listings';
```

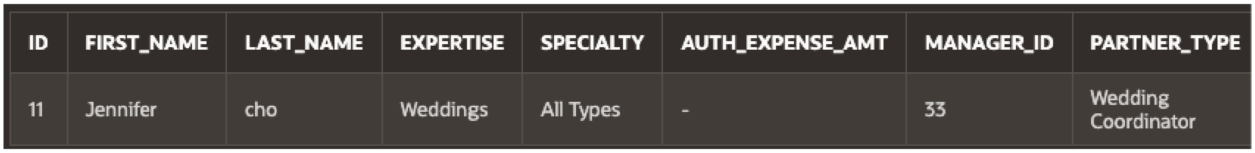


TABLE_NAME	SYNONYM_NAME
COPY_D_TRACK_LISTINGS	DJ_TRACKS

8. Create a function-based index for the last_name column in DJs on Demand D_PARTNERS table that makes it possible not to have to capitalize the table name for searches. Write a SELECT statement that would use this index.

ANS:

```
CREATE INDEX d_ptrs_last_name_idx
ON d_partners (LOWER(last_name));

SELECT *
FROM d_partners
WHERE LOWER(last_name) LIKE 'c%'
```



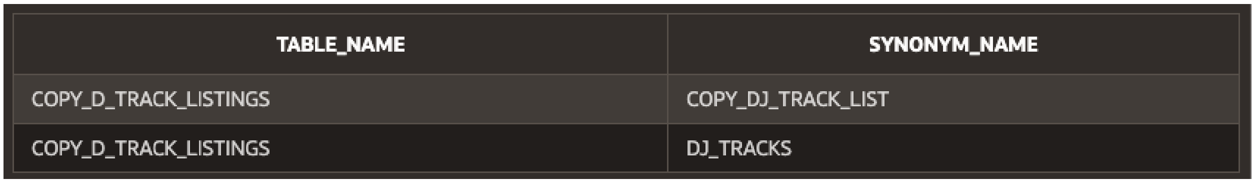
ID	FIRST_NAME	LAST_NAME	EXPERTISE	SPECIALTY	AUTH_EXPENSE_AMT	MANAGER_ID	PARTNER_TYPE
11	Jennifer	cho	Weddings	All Types	-	33	Wedding Coordinator

9. Create a synonym for the D_TRACK_LISTINGS table. Confirm that it has been created by querying the data dictionary.

ANS:

```
CREATE SYNONYM copy_dj_track_list
FOR copy_d_track_listings;
```

```
SELECT table_name, synonym_name
FROM USER_SYNONYMS
WHERE LOWER(table_name) = 'copy_d_track_listings';
```

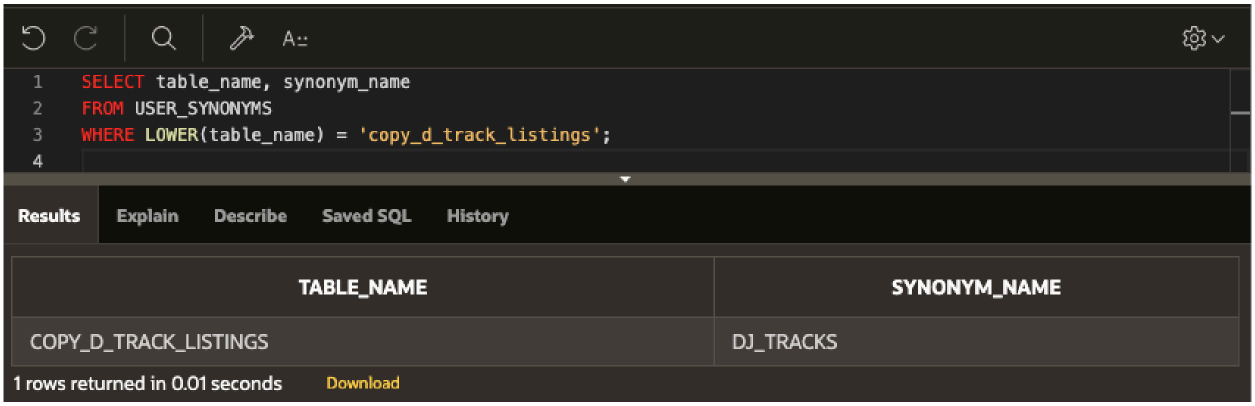


TABLE_NAME	SYNONYM_NAME
COPY_D_TRACK_LISTINGS	COPY_DJ_TRACK_LIST
COPY_D_TRACK_LISTINGS	DJ_TRACKS

10. Drop the synonym that you created in question 9.

ANS:

```
DROP SYNONYM copy_dj_track_list;
```



```sql
SELECT table_name, synonym_name
FROM USER_SYNONYMS
WHERE LOWER(table_name) = 'copy_d_track_listings';
```

TABLE_NAME	SYNONYM_NAME
COPY_D_TRACK_LISTINGS	DJ_TRACKS

1 rows returned in 0.01 seconds

17-1 Controlling User Access

NOTE: I do not have the privilege to create users or run certain commands for certain questions. Thus, some queries given as answers were not run.

1. What are system privileges concerned with?
    a. System privileges are concerned with actions at a database level, like creating tables, views, users, sequences, and sessions
2. What are object privileges concerned with?
    a. Object privileges are concerned with actions on a specific object in a database, like altering, updating, and deleting things from a specific table.
3. What is another name for object security?
    a. Data security
4. What commands are necessary to allow Scott access to the database with a password of tiger?

ANS:

```
CREATE USER Scott
IDENTIFIED BY tiger

GRANT create session
TO Scott
```

5. What are the commands to allow Scott to SELECT from and UPDATE the d_clients table?

ANS:

```
GRANT SELECT, UPDATE
   ON d_clients to Scott
```

6. What is the command to allow everybody the ability to view the d_songs table?

ANS:

```
GRANT select
   ON d_songs
TO PUBLIC
```

7. Query the data dictionary to view the object privileges granted to you the user.

ANS:

```
SELECT *
FROM USER_TAB_PRIVS_RECD;
```

```
DESCRIBE USER_TAB_PRIVS_RECD;
```

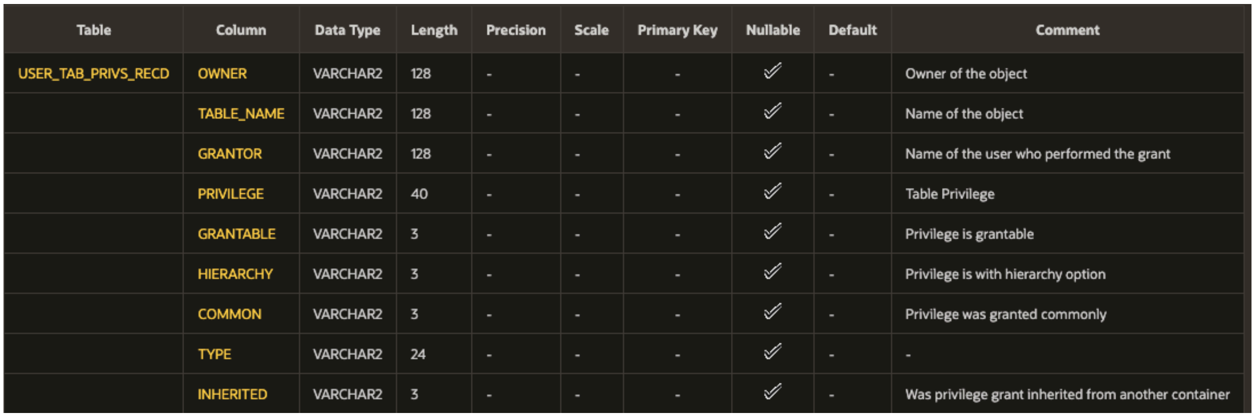


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
USER_TAB_PRIVS_RECD	OWNER	VARCHAR2	128	-	-	-	✓	-	Owner of the object
	TABLE_NAME	VARCHAR2	128	-	-	-	✓	-	Name of the object
	GRANTOR	VARCHAR2	128	-	-	-	✓	-	Name of the user who performed the grant
	PRIVILEGE	VARCHAR2	40	-	-	-	✓	-	Table Privilege
	GRANTABLE	VARCHAR2	3	-	-	-	✓	-	Privilege is grantable
	HIERARCHY	VARCHAR2	3	-	-	-	✓	-	Privilege is with hierarchy option
	COMMON	VARCHAR2	3	-	-	-	✓	-	Privilege was granted commonly
	TYPE	VARCHAR2	24	-	-	-	✓	-	-
	INHERITED	VARCHAR2	3	-	-	-	✓	-	Was privilege grant inherited from another container

8. What privilege should a user be given to create tables?
   a. create table
   b. create any table

9. If you create a table, how can you pass along privileges to other users just to view your table?

ANS:

```
GRANT select
   ON my_table_name
TO user1, user2
```

10. What syntax would you use to grant another user access to your copy_employees table?

ANS:

```
GRANT select
```

```
   ON copy_employees
TO some_user
```

11. How can you find out what privileges you have been granted for columns in the tables belonging to others?

ANS:

```
SELECT *
FROM USER_COL_PRIVS_MADE;

DESCRIBE USER_COL_PRIVS_MADE;
```

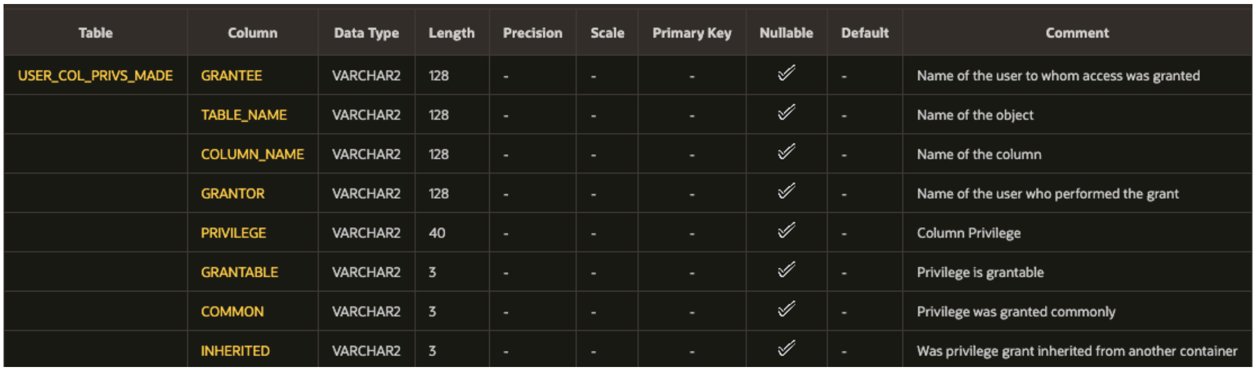


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
USER_COL_PRIVS_MADE	GRANTEE	VARCHAR2	128	-	-	-	✓	-	Name of the user to whom access was granted
	TABLE_NAME	VARCHAR2	128	-	-	-	✓	-	Name of the object
	COLUMN_NAME	VARCHAR2	128	-	-	-	✓	-	Name of the column
	GRANTOR	VARCHAR2	128	-	-	-	✓	-	Name of the user who performed the grant
	PRIVILEGE	VARCHAR2	40	-	-	-	✓	-	Column Privilege
	GRANTABLE	VARCHAR2	3	-	-	-	✓	-	Privilege is grantable
	COMMON	VARCHAR2	3	-	-	-	✓	-	Privilege was granted commonly
	INHERITED	VARCHAR2	3	-	-	-	✓	-	Was privilege grant inherited from another container

17-2 Creating and Revoking Object Privileges

1. What is a role?
    a. **Role:** a package of privileges that can be granted to user
2. What are the advantages of a role to a DBA?
    a. Easier to revoke and maintain privileges
    b. Users can be given several roles at once and several users can be assigned the same once, removing the need to grant privileges one at a time
3. Give the ability to another user in your class to look at one of your tables. Give him the right to let other students have that ability.

ANS:

```
GRANT SELECT
ON table_1
TO classmate_1
WITH GRANT OPTION
```

4. You are the DBA. You are creating many users who require the same system privileges. What should you use to make your job easier?
    a. I should create a role to bundle various system privileges that I want those users to have, allowing me to grant them all at once through that role.

5. What is the syntax to accomplish the following?
    a. Create a role of manager that has the privileges to select, insert, and update and delete from the employees table

ANS:

```
CREATE ROLE manager;

GRANT select, insert, update, delete
ON employees
TO manager;
```

   b. Create a role of clerk that just has the privileges of select and insert on the employees table

ANS:

```
CREATE ROLE clerk;

GRANT select, insert
ON employees
TO clerk;
```

   c. Grant the manager role to user scott

ANS:

```
GRANT manager
TO scott;
```

   d. Revoke the ability to delete from the employees table from the manager role

ANS:

```
REVOKE delete
ON employees
FROM manager;
```

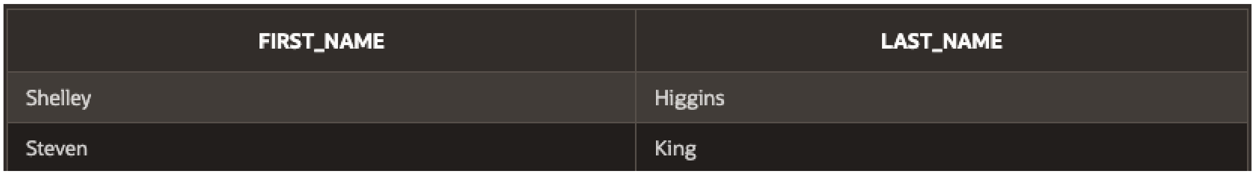
6. What is the purpose of a database link?
    a. The purpose of a database link is to allow users to access a remote database without needing to be a user of that database.

17-3 Regular Expressions

1. Working with the employees table, and using regular expressions, write a query that returns employees whose first names start with a “S” (uppercase) followed by either a “t” (lowercase) or “h” (lowercase).

ANS:

```
SELECT first_name, last_name
FROM employees
WHERE REGEXP_LIKE(first_name, '^S(t|h)');
```



FIRST_NAME	LAST_NAME
Shelley	Higgins
Steven	King

2. Investigate the LOCATIONS table.
    a. Describe the table.

ANS:

```
DESCRIBE locations
```

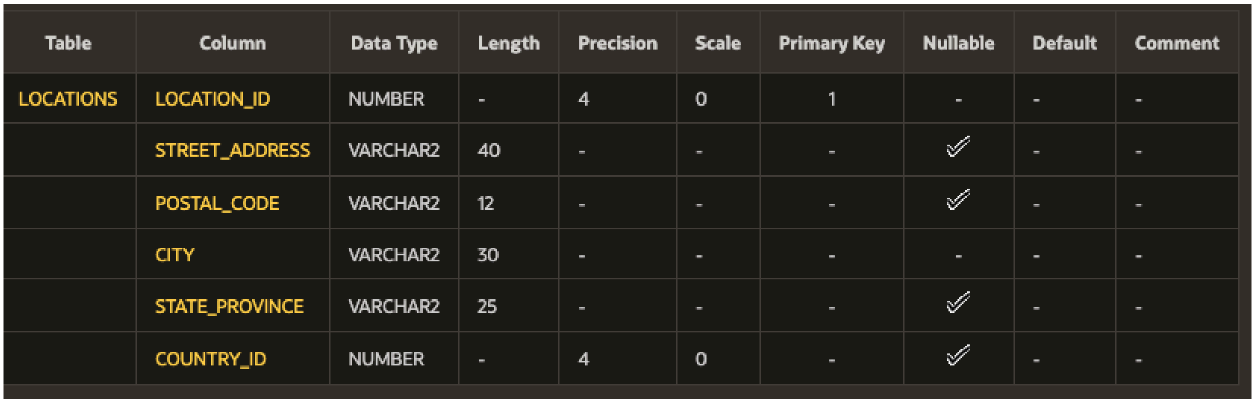
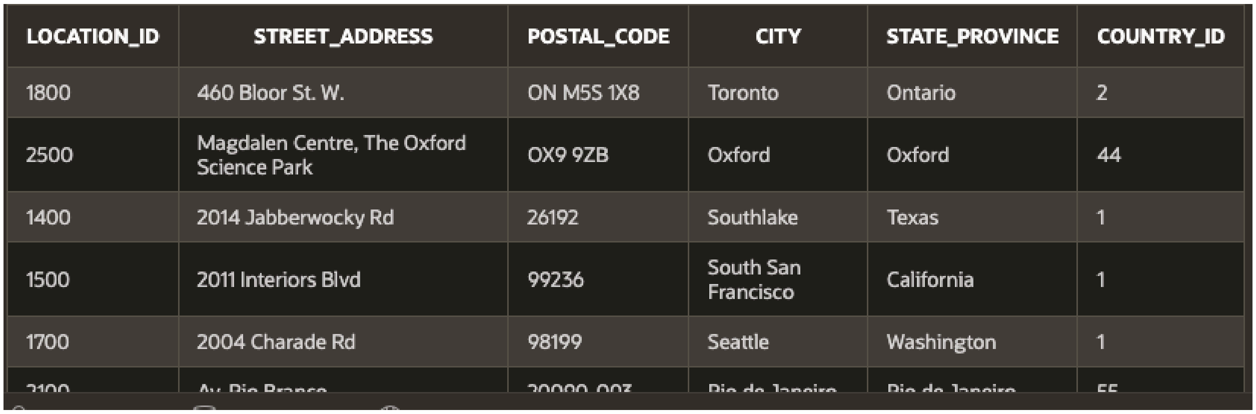


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
LOCATIONS	LOCATION_ID	NUMBER	-	4	0	1	-	-	-
	STREET_ADDRESS	VARCHAR2	40	-	-	-	✓	-	-
	POSTAL_CODE	VARCHAR2	12	-	-	-	✓	-	-
	CITY	VARCHAR2	30	-	-	-	-	-	-
	STATE_PROVINCE	VARCHAR2	25	-	-	-	✓	-	-
	COUNTRY_ID	NUMBER	-	4	0	-	✓	-	-

    b. Perform a select that returns all rows and all columns of that table.

ANS:

```
SELECT *  FROM locations
```

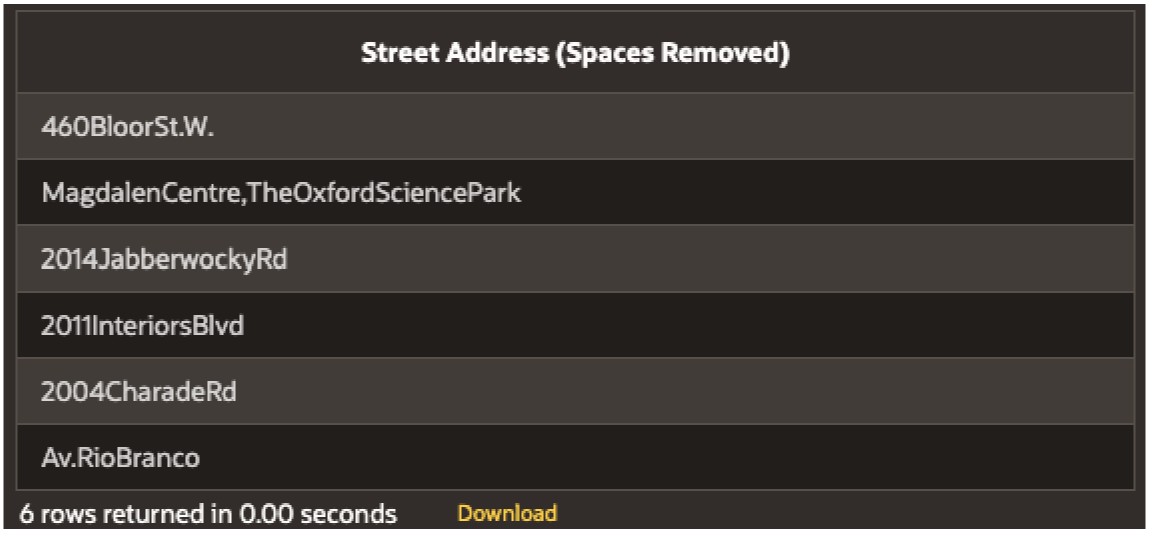


LOCATION_ID	STREET_ADDRESS	POSTAL_CODE	CITY	STATE_PROVINCE	COUNTRY_ID
1800	460 Bloor St. W.	ON M5S 1X8	Toronto	Ontario	2
2500	Magdalen Centre, The Oxford Science Park	OX9 9ZB	Oxford	Oxford	44
1400	2014 Jabberwocky Rd	26192	Southlake	Texas	1
1500	2011 Interiors Blvd	99236	South San Francisco	California	1
1700	2004 Charade Rd	98199	Seattle	Washington	1
2100	Av. Rio Branco	20090-003	Rio de Janeiro	Rio de Janeiro	55

c. Write a query using regular expressions that removes the spaces in the street_address column in the LOCATIONS table.

ANS:

SELECT REGEXP_REPLACE(street_address, ' ', '') "Street Address (Spaces Removed)"
FROM locations



Street Address (Spaces Removed)
460BloorSt.W.
MagdalenCentre,TheOxfordSciencePark
2014JabberwockyRd
2011InteriorsBlvd
2004CharadeRd
Av.RioBranco

6 rows returned in 0.00 seconds