Database Programming with SQL
14-1: Intro to Constraints; NOT NULL and UNIQUE Constraints
Practice Activities

Objectives

- Define the term "constraint" as it relates to data integrity
- State when it is possible to define a constraint at the column level, and when it is possible at the table level
- State why it is important to give meaningful names to constraints
- State which data integrity rules are enforced by NOT NULL and UNIQUE constraints
- Write a CREATE TABLE statement which includes NOT NULL and UNIQUE constraints at the table and column levels
- Explain how constraints are created at the time of table creation

Vocabulary

Identify the vocabulary word for each definition below.

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	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
FOREIGN KEY constraint	Identifies that table and column in the parent table
UNIQUE constraint	An integrity constraint that requires every value in a column or set of columns be unique
FOREIGN KEY constraint	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

Try It / Solve It

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	NUMBER	6			not null	
name	VARCHAR2	50			nullable	
date_opened	DATE				not null	
address	VARCHAR2	100			not null	
city	VARCHAR2	50			not null	
zip/postal code	VARCHAR2	10			nullable	
phone	VARCHAR2	20			nullable	
email	VARCHAR2	80			not null	
manager_id	NUMBER	6			nullable	
Emergency contact	VARCHAR2	20			nullable	

1. What is a "constraint" as it relates to data integrity?

   A constraint is a database rule used to enforce data integrity. It ensures the accuracy and validity of data entered into the database by preventing invalid entries.

2. What are the limitations of constraints that may be applied at the column level and at the table level?

   Column-level constraints: Apply to a single column. Suitable for simpler constraints like NOT NULL or UNIQUE.
   Table-level constraints: Can reference multiple columns (composite keys).Used for constraints like PRIMARY KEY, FOREIGN KEY, or composite UNIQUE constraints.

3. Why is it important to give meaningful names to constraints?

   Meaningful constraint names make it easier to identify and manage constraints in the database, reducing confusion and improving maintainability.

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.

   See above table.

5. Use "nullable" to indicate those columns that can have null values.

   See above table.

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

```
CREATE TABLE global_locations (
    id NUMBER(6) CONSTRAINT global_locations_id_pk PRIMARY KEY,
    name VARCHAR2(50),
    date_opened DATE NOT NULL,
    address VARCHAR2(100) NOT NULL,
    city VARCHAR2(50) NOT NULL,
    zip_postal_code VARCHAR2(10),
    phone VARCHAR2(20),
    email VARCHAR2(80),
    manager_id NUMBER(6),
    emergency_contact VARCHAR2(20),
    CONSTRAINT global_locations_email_uk UNIQUE (email)
);
```

CREATE TABLE global_locations (
    id NUMBER(6) CONSTRAINT global_locations_id_pk PRIMARY KEY,
    name VARCHAR2(50),
    date_opened DATE NOT NULL,
    address VARCHAR2(100) NOT NULL,
    city VARCHAR2(50) NOT NULL,
    zip_postal_code VARCHAR2(10),
    phone VARCHAR2(20),
    email VARCHAR2(80),
    manager_id NUMBER(6),
    emergency_contact VARCHAR2(20),
    CONSTRAINT global_locations_email_uk UNIQUE (email)
);

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

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

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.

ORACLE
Academy

academy.oracle.com

Database Programming with SQL
14-2: PRIMARY KEY, FOREIGN KEY, and CHECK Constraints
Practice Activities

Objectives

- Define and give an example of PRIMARY KEY, FOREIGN KEY, and CHECK constraints
- Explain the purpose of defining PRIMARY KEY, FOREIGN KEY, and CHECK constraints on a table
- Demonstrate the creation of constraints at the column level and table level in a CREATE TABLE statement
- Evaluate a business problem requiring the addition of a PRIMARY KEY and FOREIGN KEY constraint and write the code to execute the change

Vocabulary

Identify the vocabulary word for each definition below.

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	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	Establishes a relationship between the foreign key column and a primary key or unique key in the same table or a different table

Try It / Solve It

1. What is the purpose of a
   a. PRIMARY KEY
   b. FOREIGN KEY
   c. CHECK CONSTRAINT

a. **PRIMARY KEY:** Ensures that each row in a table is uniquely identifiable by disallowing duplicate and null values.
b. **FOREIGN KEY:** Establishes a relationship between tables by referencing the PRIMARY KEY of another table, ensuring referential integrity.
c. **CHECK CONSTRAINT:** Ensures that specific conditions or rules are met for a column's values.

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

Column Name	**Constraint**
animal_id	PRIMARY KEY
license_tag_number	UNIQUE
admit_date	NOT NULL
vaccination_date	NOT NULL

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

```
CREATE TABLE animals (
    animal_id NUMBER(6) CONSTRAINT animals_animal_id_pk PRIMARY KEY,
    name VARCHAR2(25),
    license_tag_number NUMBER(10) CONSTRAINT animals_license_tag_uk UNIQUE,
    admit_date DATE CONSTRAINT animals_admit_date_nn NOT NULL,
    adoption_id NUMBER(5),
    vaccination_date DATE CONSTRAINT animals_vaccination_date_nn NOT NULL
);
```

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

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.

Column-level:

```
CREATE TABLE animals (
    ...
    adoption_id NUMBER(5) CONSTRAINT animals_adoption_fk REFERENCES adoptions(adoption_id)
);
```

Table-level:

```
CREATE TABLE animals (
    ...
    adoption_id NUMBER(5),
    CONSTRAINT animals_adoption_fk FOREIGN KEY (adoption_id) REFERENCES adoptions(adoption_id)
);
```

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

a. ON DELETE CASCADE: If a parent row is deleted, all child rows referencing it are also deleted automatically.
b. ON DELETE SET NULL: If a parent row is deleted, the child rows retain their presence, but the FOREIGN KEY column is set to NULL.

7. What are the restrictions on defining a CHECK constraint?

The CHECK condition must only reference the current row.
Cannot use functions like SYSDATE, USER, or pseudocolumns such as CURRVAL or NEXTVAL.
Cannot reference columns from other tables.

ORACLE Academy

academy.oracle.com

Database Programming with SQL
14-3: Managing Constraints
Practice Activities

Objectives

- List four different functions that the ALTER statement can perform on constraints
- Write ALTER TABLE statements to add, drop, disable, and enable constraints
- Name a business function that would require a DBA to drop, enable, and/or disable a constraint or use the CASCADE syntax
- Query the data dictionary for USER_CONSTRAINTS and interpret the information returned

Vocabulary

Identify the vocabulary word for each definition below.

DISABLE CONSTRAINT	To deactivate an integrity constraint
CASCADE	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 CONSTRAINTS	Defines the actions the database server takes when a user attempts to delete or update a key to which existing foreign keys point

Try It / Solve It

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?

```
Add a constraint
Drop a constraint
Enable a constraint
Disable a constraint
```

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?

```
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?

```
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 tablenames must be capitalized.

   a. The constraint name for the primary key in the copy_d_clients table is COPY_D_CLIENTS_PK.

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

5. Drop the PRIMARY KEY constraint on the copy_d_clients table. Explain your results.

   Error. Cannot drop PK because of FK dependency. Need to use CASCADE.

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

Error at line 1/13: ORA-00913: too many values. The ORA-00913: too many values error occurs because the number of values provided in the INSERT statement exceeds the number of columns in the copy_d_events table.

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.

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

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

```
INSERT INTO copy_d_events VALUES
        (140, 'Cline Bas Mitzvah', TO_DATE('15-Jul-2004', 'DD-MON-YYYY'), 'Church and Private Home', 4500,
        105, 87, 77, 7125
        );
```

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

   We get the error ORA-02430: cannot enable constraint (COPY_D_CLIENTS_PK) - no such constraint. This is because we dropped the constraint. When the constraint is dropped, it is completely removed from the table's metadata.

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

    You must ensure that all client_number values in copy_d_events exist in copy_d_clients. Then:

```
ALTER TABLE copy_d_events
ENABLE CONSTRAINT copy_d_events_fk;
```

11. Why might you want to disable and then re-enable a constraint?

    To improve performance during bulk inserts/updates.
    To correct or clean up invalid data before enforcing constraints.

12. Query the data dictionary for some of the constraints that you have created. How does the data dictionary identify each constraint type?

```
SELECT constraint_name, table_name, constraint_type
FROM USER_CONSTRAINTS
WHERE table_name IN ('COPY_D_CLIENTS', 'COPY_D_EVENTS');
```

The constraint_type column identifies constraint types:
P: PRIMARY KEY
R: FOREIGN KEY
U: UNIQUE
C: CHECK (includes NOT NULL)

ORACLE Academy

academy.oracle.com

Database Programming with SQL
15-1: Creating Views
Practice Activities

Objectives

- List three uses for views from the standpoint of a database administrator
- Explain, from a business perspective, why it is important to be able to create and use logical subsets of data derived from one or more tables
- Create a view with and without column aliases in the subquery using a single base table
- Create a complex view that contains group functions to display values from two tables
- Retrieve data from a view

Vocabulary

Identify the vocabulary word for each definition below.

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

Try It / Solve It

1. What are three uses for a view from a DBA's perspective?

   Restrict access to specific columns or rows in a table for security purposes.
   Simplify complex queries by encapsulating them in a view.
   Present data in different formats for different users without altering the underlying tables.

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.

```
CREATE OR REPLACE VIEW view_d_songs AS
SELECT ID,
       TITLE AS "Song Title",
       ARTIST
FROM d_songs
WHERE TYPE_CODE = 77;
```

3. SELECT *
   FROM view_d_songs.

   What was returned?

```
 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.

```
CREATE OR REPLACE VIEW view_d_songs AS
SELECT ID AS "ID",
       TITLE AS "Song Title",
       ARTIST AS "Artist",
       TYPE_CODE AS "Type Code"
FROM d_songs
WHERE TYPE_CODE = 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.

```
CREATE OR REPLACE VIEW view_d_events AS
SELECT NAME AS "Event Name",
       EVENT_DATE AS "Event Date",
       DESCRIPTION AS "Theme Description"
FROM d_events;
```

```
Event Name         Event Date         Theme Description
Peters Graduation  05/14/2004         Party for 200, red, white, blue motif
Vigil wedding      04/28/2004         Black tie at Four Season hotel
```

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.

```
CREATE OR REPLACE VIEW view_department_salary_summary AS
SELECT DEPARTMENT_ID AS "Department ID",
       MIN(SALARY) AS "Minimum Salary",
       MAX(SALARY) AS "Maximum Salary",
       AVG(SALARY) AS "Average Salary"
FROM EMPLOYEES
GROUP BY DEPARTMENT_ID
ORDER BY 1;
```

Database Programming with SQL
15-2: DML Operations and Views
Practice Activities

Objectives

- Write and execute a query that performs DML operations on a simple view
- Name the conditions that restrict modifying a view using DML operations
- Write and execute a query using the WITH CHECK OPTION clause
- Explain the use of WITH CHECK OPTION as it applies to integrity constraints and data validation
- Apply the WITH READ ONLY option to a view to restrict DML operations

Vocabulary

Identify the vocabulary word for each definition below.

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

Try It / Solve It

Use the DESCRIBE statement to verify that you have tables named copy_d_songs, copy_d_events, copy_d_cds, and copy_d_clients in your schema. If you don't, write a query to create a copy of each.

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.

```
SELECT TABLE_NAME, COLUMN_NAME, UPDATABLE, INSERTABLE, DELETABLE
FROM USER_UPDATABLE_COLUMNS
WHERE TABLE_NAME IN ('COPY_D_SONGS', 'COPY_D_EVENTS', 'COPY_D_CDS', 'COPY_D_CLIENTS');
```

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.

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

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

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.

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

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

```
DELETE FROM read_copy_d_cds
WHERE CD_NUMBER = 90;
```

```
Error at line 1/13: ORA-42399: cannot perform a DML operation on a
read-only view

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

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

```
CREATE OR REPLACE VIEW read_copy_d_cds AS
SELECT *
FROM copy_d_cds
WHERE YEAR = 2000
WITH CHECK OPTION CONSTRAINT ck_read_copy_d_cds;
```

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

```
DELETE FROM read_copy_d_cds
WHERE YEAR = 2000;
```

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

```
DELETE FROM read_copy_d_cds
WHERE CD_NUMBER = 90;
```

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

```
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?

```
SELECT *
FROM copy_d_cds;
```

Only the records where YEAR = 2000 were deleted.

11. What are the restrictions on modifying data through a view?

DML restrictions apply if:
The view contains GROUP BY, DISTINCT, or aggregate functions.
The view includes WITH READ ONLY.
The view omits NOT NULL columns from the base table.
WITH CHECK OPTION enforces that changes must satisfy the view's WHERE clause.

12. What is Moore's Law? Do you consider that it will continue to apply indefinitely? Support your opinion with research from the internet.

Moore's Law predicts that the number of transistors on a chip will double approximately every two years, leading to exponential growth in computing power.
Current Relevance: Recent advancements in chip manufacturing suggest that Moore's Law is slowing as physical limitations (e.g., transistor size) are approached.
Research Insight: Technologies like quantum computing and new materials (e.g., graphene) may drive future performance improvements beyond traditional silicon-based chips.

13. What is the "singularity" in terms of computing?

The "singularity" refers to a hypothetical point in the future when artificial intelligence surpasses human intelligence, leading to rapid and unpredictable technological advancements.

Database Programming with SQL
15-3: Managing Views
Practice Activities

Objectives

- Create and execute a query that removes a view
- Create and execute a query using an inline view
- Create and execute a top-n-analysis query

Vocabulary

Identify the vocabulary word for each definition below.

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

Try It / Solve It

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.

```
CREATE OR REPLACE VIEW view_copy_d_songs AS
SELECT TITLE, ARTIST
FROM copy_d_songs;
```

```
select * from view_copy_d_songs
```

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

```
DROP VIEW view_copy_d_songs;
```

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.

```
SELECT ROWNUM AS "Rank", LAST_NAME, SALARY
FROM (SELECT LAST_NAME, SALARY
      FROM employees
      ORDER BY SALARY DESC)
WHERE ROWNUM <= 3;
```

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.

```
SELECT e.LAST_NAME, e.SALARY, e.DEPARTMENT_ID, d.MAX_SALARY
FROM employees e,
     (SELECT DEPARTMENT_ID, MAX(SALARY) AS MAX_SALARY
      FROM employees
      GROUP BY DEPARTMENT_ID) d
WHERE e.DEPARTMENT_ID = d.DEPARTMENT_ID
AND e.SALARY = d.MAX_SALARY;
```

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

```
SELECT ROWNUM AS "Rank", LAST_NAME, SALARY
FROM (SELECT LAST_NAME, SALARY
      FROM employees
      ORDER BY SALARY ASC);
```

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.

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.

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

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

department_id	department_name
105	Advertising
120	Custodial
130	Planning

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

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

7. Add the Human Resources department, department ID 220, to my_departments using view_my_departments.

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

   See chart below

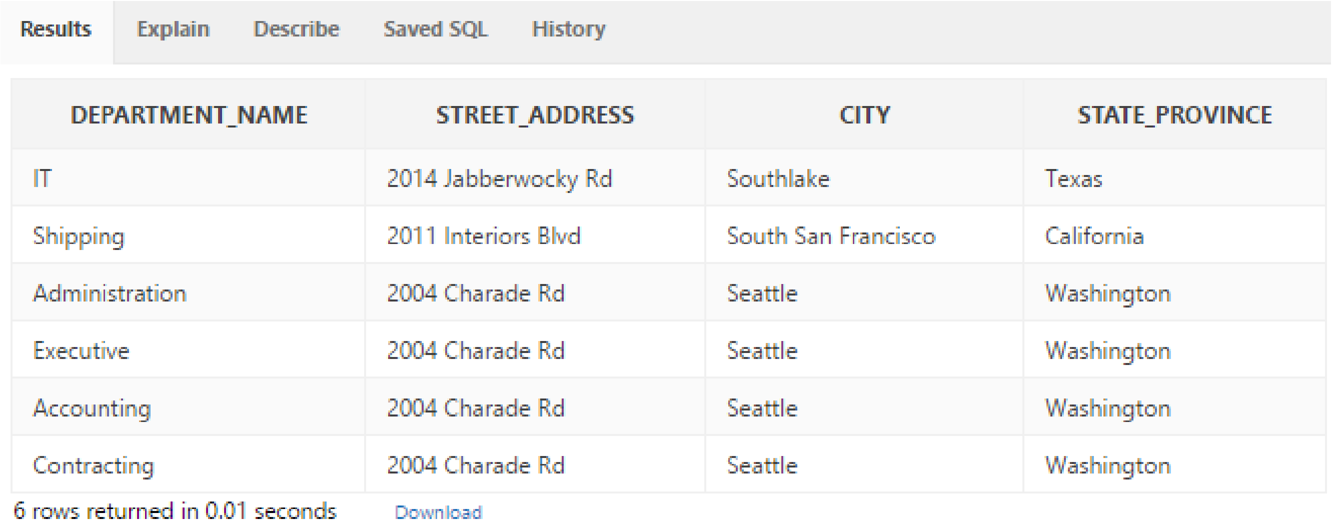
9. Modify view_my_departments to include location ID. Do a SELECT * command to show what columns are present and a DESCRIBE command to view the columns and associated constraints.

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

   See chart below

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800

Results of select statement from view:



DEPARTMENT_NAME	STREET_ADDRESS	CITY	STATE_PROVINCE
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 0.01 seconds