Vanessa Chen
HHA 507 - Oracle SQL Training
Dr. Hants Williams

Database Programming with SQL 12-1: INSERT Statements Practice Activities

- *Vocabulary:* Identify the vocabulary word for each definition below
  - Someone doing "real work" with the computer, using it as a means rather than an end
    - Answer: USER
  - Consists of a collection of DML statements that form a logical unit of work
    - Answer: Transaction
  - Fully and clearly expressed; leaving nothing implied
    - Answer: Explicit
  - Adds a new row to a table
    - Answer: INSERT
- *Try It / Solve It*
  - 1. Give two examples of why it is important to be able to alter the data in a database
    - 1. It is important to alter the data in a database for any corrections or updates because you want your data to be accurate and up-to-date
    - 2. It is important to alter the data in a database for adding any new information because you want your data to be accurate and up-to-date
  - 2. DJs on Demand just purchased four new CDs. Use an explicit INSERT statement to add each CD to the copy_d_cds table. After completing the entries, execute a SELECT * statement to verify your work

CD_Number	Title	Producer	Year
97	Celebrate the Day	R & B Inc.	2003
98	Holiday Tunes for All Ages	Tunes are Us	2004
99	Party Music	Old Town Records	2004
100	Best of Rock and Roll	Old Town Records	2004

    - INSERT INTO copy_d_cds (CD_Number, Title, Producer, Year) VALUES (97, 'Celebrate the Day', 'R & B Inc.', TO_DATE('2003', 'YYYY'));
    - INSERT INTO copy_d_cds (CD_Number, Title, Producer, Year) VALUES (98, 'Holiday Tunes for All Ages', 'Tunes are Us', TO_DATE('2004', 'YYYY'));
    - INSERT INTO copy_d_cds (CD_Number, Title, Producer, Year) VALUES (99, 'Party Music', 'Old Town Records', TO_DATE('2004', 'YYYY'));
    - INSERT INTO copy_d_cds (CD_Number, Title, Producer, Year) VALUES (100, 'Best of Rock and Roll', 'Old Town Records', TO_DATE('2004', 'YYYY'));

- SELECT * FROM copy_d_cds;

- 3. DJs on Demand has two new events coming up. One event is a fall football party and the other event is a sixties theme party. The DJs on Demand clients requested the songs shown in the table for their events. Add these songs to the copy_d_songs table using an implicit INSERT statement

ID	Title	Duration	Type_Code
52	Surfing Summer	Not known	12
53	Victory Victory	5 min	12

  - INSERT INTO copy_d_songs (ID, Title, Duration, Type_Code)
    VALUES (52, 'Surfing Summer', 'Not known', 12);
  - INSERT INTO copy_d_songs (ID, Title, Duration, Type_Code)
    VALUES (53, 'Victory Victory', '5 min', 12);
  - SELECT * FROM copy_d_songs;
- 4. Add the two new clients to the copy_d_clients table. Use either an implicit or an explicit INSERT

Client_Number	First_Name	Last_Name	Phone	Email
6655	Ayako	Dahish	3608859030	dahisha@harbor.net
6689	Nick	Neuville	9048953049	nnicky@charter.net

  - INSERT INTO copy_d_clients (Client_Number, First_Name, Last_Name, Phone, Email)
    VALUES (6655, 'Ayako', 'Dahish', '3608859030', 'dahisha@harbor.net');
  - INSERT INTO copy_d_clients (Client_Number, First_Name, Last_Name, Phone, Email)
    VALUES (6689, 'Nick', 'Neuville', '9048953049', 'nnicky@charter.net');
- 5. Add the new client's events to the copy_d_events table. The cost of each event has not been determined at this date

ID	Name	Event_ Date	Description	Cost	Venue_ ID	Package_ Code	Theme_ Code	Client_ Number
110	Ayako Anniversary	07-Jul-2004	Party for 50, sixties dress, decorations		245	79	240	6655
115	Neuville Sports Banquet	09-Sep-2004	Barbecue at residence, college alumni, 100 people		315	87	340	6689

  - INSERT INTO copy_d_events (ID, Name, Event_Date, Description, Cost, Venue_ID, Package_Code, Theme_Code, Client_Number)
    VALUES (110, 'Ayako Anniversary', TO_DATE('07-Jul-2004', 'DD-Mon-YYYY'), 'Party for 50, sixties dress, decorations', '', 245, 79, 240, 6655);
  - INSERT INTO copy_d_events (ID, Name, Event_Date, Description, Cost, Venue_ID, Package_Code, Theme_Code, Client_Number)
    VALUES (115, 'Neuville Sports Banquet', TO_DATE('09-Sep-2004', 'DD-Mon-YYYY'), 'Barbecue at residence, collect alumni, 100 people', '', 315, 87, 340, 6689);

- 6. Create a table called rep_email using the following statement. Populate this table by running a query on the employees table that includes only those employees who are REP's
  - CREATE TABLE rep_email (
    id NUMBER(3) CONSTRAINT rel_id_pk PRIMARY KEY,
    first_name VARCHAR2(10),
    last_name VARCHAR2(10),
    email_address VARCHAR2(10))
    - INSERT INTO rep_email (id, first_name, last_name, email_address)
      SELECT employee_id, first_name, last_name, email
      FROM employees
      WHERE job_title = 'REP';

Database Programming with SQL 12-2: Updating Column Values and Deleting Rows Practice Activities

- *Vocabulary:* Identify the vocabulary word for each definition below
  - Modifies existing rows in a table
    - Answer: UPDATE
  - Retrieves information from one table & uses the information to update another table
    - Answer: Correlated subquery UPDATE
  - Ensures that the data adheres to a predefined set of rules
    - Answer: Integrity constraint
  - Deletes information on a linked table based on what was deleted on the other table
    - Answer: Correlated subquery DELETE
  - Removes existing rows from a table
    - Answer: DELETE
- *Try It / Solve It*
  - NOTE: Copy tables in this section do not exist
    - If any change is not possible, give an explanation as to why it is not possible
  - 1. Monique Tuttle, the manager of Global Fast Foods, sent a memo requesting an immediate change in prices. The price for a strawberry shake will be raised from $3.59 to $3.75, and the price for fries will increase to $1.20. Make these changes to the copy_f_food_items table
    - UPDATE copy_f_food_items
      SET price = 3.75
      WHERE item_name = 'Strawberry Shake';
    - UPDATE copy_f_food_items

SET price = 1.20
WHERE item_name = 'Fries';

- 2. Bob Miller and Sue Doe have been outstanding employees at Global Fast Foods. Management has decided to reward them by increasing their overtime pay. Bob Miller will receive an additional $0.75 per hour and Sue Doe will receive an additional $0.85 per hour. Update the copy_f_staffs table to show these new values. (Note: Bob Miller currently doesn't get overtime pay. What function do you need to use to convert a null value to 0?)
  - UPDATE copy_f_staffs
    SET overtime_pay = COALESCE(overtime_pay, 0) + 0.75
    WHERE staff_name = 'Bob Miller';
  - UPDATE copy_f_staffs
    SET overtime_pay = COALESCE(overtime_pay, 0) + 0.85
    WHERE staff_name = 'Sue Doe';
- 3. Add the orders shown to the Global Fast Foods copy_f_orders table

ORDER_NUMBER	ORDER_DATE	ORDER_TOTAL	CUST_ID	STAFF_ID
5680	June 12, 2004	159.78	145	9
5691	09-23-2004	145.98	225	12
5701	July 4, 2004	229.31	230	12

  - INSERT INTO copy_f_orders (ORDER_NUMBER, ORDER_DATE, ORDER_TOTAL, CUST_ID, STAFF_ID)
    VALUES (5680, TO_DATE('June 12, 2004', 'Month DD, YYYY'), 159.80, 145, 9);
  - INSERT INTO copy_f_orders (ORDER_NUMBER, ORDER_DATE, ORDER_TOTAL, CUST_ID, STAFF_ID)
    VALUES (5691, TO_DATE('09-23-2004', 'MM-DD-YYYY'), 145.98, 225, 12);
  - INSERT INTO copy_f_orders (ORDER_NUMBER, ORDER_DATE, ORDER_TOTAL, CUST_ID, STAFF_ID)
    VALUES (5701, TO_DATE('July 4, 2004', 'Month DD, YYYY'), 229.31, 230, 12);
- 4. Add the new customers shown below to the copy_f_customers table. You may already have added Katie Hernandez. Will you be able to add all these records successfully?

ID	FIRST_ NAME	LAST_ NAME	ADDRESS	CITY	STATE	ZIP	PHONE_NUMBER
145	Katie	Hernandez	92 Chico Way	Los Angeles	CA	98008	8586667641
225	Daniel	Spode	1923 Silverado	Denver	CO	80219	7193343523
230	Adam	Zurn	5 Admiral Way	Seattle	WA		4258879009

  - INSERT INTO copy_f_customers (ID, FIRST_NAME, LAST_NAME, ADDRESS, CITY, STATE, ZIP, PHONE_NUMBER)
    VALUES (145, 'Katie', 'Hernandez', '92 Chico Way', 'Los Angeles', 'CA', '98008', '85866667641');

        - INSERT INTO copy_f_customers (ID, FIRST_NAME, LAST_NAME, ADDRESS, CITY, STATE, ZIP, PHONE_NUMBER)
          VALUES (225, ‘Daniel’, ‘Spode’, ‘1923 Silverado’, ‘Denver’, ‘CO’, ‘80219’, ‘7193343523’);
        - INSERT INTO copy_f_customers (ID, FIRST_NAME, LAST_NAME, ADDRESS, CITY, STATE, ZIP, PHONE_NUMBER)
          VALUES (230, ‘Adam’, ‘Zurn’, ‘5 Admiral Way’, ‘Seattle’, ‘WA’, ‘’, ‘4258879009’);
    - 5. Sue Doe has been an outstanding Global Foods staff member and has been given a salary raise. She will now be paid the same as Bob Miller. Update her record in copy_f_staffs
        - UPDATE copy_f_staffs
          SET salary = (SELECT salary FROM copy_f_staffs WHERE staff_name = 'Bob Miller')
          WHERE staff_name = 'Sue Doe';
    - 6. Global Fast Foods is expanding their staff. The manager, Monique Tuttle, has hired Kai Kim. Not all information is available at this time, but add the information shown here

ID	FIRST_NAME	LAST_NAME	BIRTHDATE	SALARY	STAFF_TYPE
25	Kai	Kim	3-Nov-1988	6.75	Order Taker

        - INSERT INTO copy_f_staffs (ID, FIRST_NAME, LAST_NAME, BIRTHDATE, SALARY, STAFF_TYPE)
          VALUES (25, ‘Kai’, ‘Kim’, TO_DATE(‘3-Nov-1988’, ‘DD-Mon-YYYY’), 6.75, ‘Order Taker’);
    - 7. Now that all the information is available for Kai Kim, update his Global Fast Foods record to include the following: Kai will have the same manager as Sue Doe. He does not qualify for overtime. Leave the values for training, manager budget, and manager target as null
        - UPDATE copy_f_staffs
        - SET manager_id = (SELECT manager_id FROM copy_f_staffs WHERE staff_name = 'Sue Doe'),
            overtime_eligible = 'No',
            training = NULL,
            manager_budget = NULL,
            manager_target = NULL
          WHERE ID = 25;
    - 8. Execute the following SQL statement. Record your results
        - DELETE from departments
          WHERE department_id = 60;
        - Answer: ORA-02292: integrity constraint (US_A927_SQL_S02.EMP_DEPT_FK) violated - child record found

- 9. Kim Kai has decided to go back to college and does not have the time to work and go to school. Delete him from the Global Fast Foods staff. Verify that the change was made
  - DELETE FROM copy_f_staffs
    WHERE ID = 25;
  - SELECT *
    FROM copy_f_staffs
    WHERE ID = 25;
- 10. Create a copy of the employees table and call it lesson7_emp; Once this table exists, write a correlated delete statement that will delete any employees from the lesson7_employees table that also exist in the job_history table
  - CREATE TABLE lesson7_emp AS
    SELECT *
    FROM employees;

    DELETE FROM lesson7_emp e
    WHERE EXISTS (
      SELECT 1 FROM job_history j
      WHERE j.employee_id = e.employee_id
    );

Database Programming with SQL 12-3: DEFAULT Values, MERGE, and Multi-Table Inserts Practice Activities

- *Try It / Solve It*
  - 1. When would you want a DEFAULT value?
    - You would want a DEFAULT value when a specific value should be automatically filled unless a different value is explicitly stated during the data entry
  - 2. Currently, the Global Foods F_PROMOTIONAL_MENUS table START_DATE column does not have SYSDATE set as DEFAULT. Your manager has decided she would like to be able to set the starting date of promotions to the current day for some entries. This will require three steps:
    - a. In your schema, Make a copy of the Global Foods F_PROMOTIONAL_MENUS table using the following SQL statement:
      - CREATE TABLE copy_f_promotional_menus AS (SELECT * FROM f_promotional_menus)
    - b. Alter the current START_DATE column attributes using:
      - ALTER TABLE copy_f_promotional_menus MODIFY(start_date DATE DEFAULT SYSDATE)

- c. INSERT the new information and check to verify the results. INSERT a new row into the copy_f_promotional_menus table for the manager's new promotion. The promotion code is 120. The name of the promotion is 'New Customer.' Enter DEFAULT for the start date and '01-Jun-2005' for the ending date. The giveaway is a 10% discount coupon. What was the correct syntax used?
  - INSERT INTO copy_f_promotional_menus (promo_code, promo_name, start_date, end_date, giveaway)
    VALUES (120, 'New Customer', DEFAULT,
    TO_DATE('01-Jun-2005', 'DD-Mon-YYYY'), '10% Discount Coupon');
  - SELECT * FROM copy_f_promotional_menus
    WHERE promo_code = 120;

- 3. Allison Plumb, the event planning manager for DJs on Demand, has just given you the following list of CDs she acquired from a company going out of business. She wants a new updated list of CDs in inventory in an hour, but she doesn't want the original D_CDS table changed. Prepare an updated inventory list just for her
  - a. Assign new cd_numbers to each new CD acquired
  - b. Create a copy of the D_CDS table called manager_copy_d_cds. What was the correct syntax used?
    - CREATE TABLE manager_copy_d_cds AS
      SELECT * FROM D_CDS WHERE 1=2;
  - c. INSERT into the manager_copy_d_cds table each new CD title using an INSERT statement. Make up one example or use this data: 20, 'Hello World Here I Am', 'Middle Earth Records', '1998' What was the correct syntax used?
    - INSERT INTO manager_copy_d_cds (cd_number, cd_title, record_label, release_year)
      VALUES (20, 'Hello World Here I Am', 'Middle Earth Records', TO_DATE('1998', 'YYYY');
  - d. Use a merge statement to add to the manager_copy_d_cds table, the CDs from the original table. If there is a match, update the title and year. If not, insert the data from the original table. What was the correct syntax used?
    - MERGE INTO manager_copy_d_cds t
      USING D_CDS s
      ON (t.cd_number = s.cd_number)
      WHEN MATCHED THEN
        UPDATE SET t.cd_title = s.cd_title, t.release_year = s.release_year

```
    WHEN NOT MATCHED THEN
      INSERT (cd_number, cd_title, record_label, release_year)
      VALUES (s.cd_number, s.cd_title, s.record_label,
    s.release_year);
```

- 4. Run the following 3 statements to create 3 new tables for use in a Multi-table insert statement. All 3 tables should be empty on creation, hence the WHERE 1=2 condition in the WHERE clause. Once the tables exist in your account, write a Multi-Table insert statement to first select the employee_id, hire_date, salary, and manager_id of all employees. If the salary is more than 20000 insert the employee_id and salary into the special_sal table. Insert the details of employee_id, hire_date, and salary into the sal_history table. Insert the employee_id, manager_id, and salary into the mgr_history table. You should get a message back saying 39 rows were inserted. Verify you get this message and verify you have the following number of rows in each table: Sal_history = 19 rows, Mgr_history = 19 rows, Special_sal = 1
  - CREATE TABLE sal_history (employee_id, hire_date, salary)
    AS SELECT employee_id, hire_date, salary
    FROM employees
    WHERE 1=2;
  - CREATE TABLE mgr_history (employee_id, manager_id, salary)
    AS SELECT employee_id, manager_id, salary
    FROM employees
    WHERE 1=2;
  - CREATE TABLE special_sal (employee_id, salary)
    AS SELECT employee_id, salary
    FROM employees
    WHERE 1=2;

```
INSERT ALL
 INTO special_sal (employee_id, salary)
  SELECT employee_id, salary
  FROM employees
  WHERE salary > 20000
 INTO sal_history (employee_id, hire_date, salary)
  SELECT employee_id, hire_date, salary
  FROM employees
  WHERE salary > 20000
 INTO mgr_history (employee_id, manager_id, salary)
  SELECT employee_id, manager_id, salary
  FROM employees
  WHERE salary > 20000
```

SELECT * FROM dual;
    - SELECT COUNT(*) AS sal_history_count
      FROM sal_history;
    - SELECT COUNT(*) AS mgr_history_count
      FROM mgr_history;
    - SELECT COUNT(*) AS special_sal_count
      FROM special_sal;

Database Programming with SQL 13-1: Creating Tables Practice Activities

- *Vocabulary:* Identify the vocabulary word for each definition below
    - Created and maintained by the Oracle Server and contains information about the database
        - Answer: Data dictionary
    - A collection of objects that are the logical structures that directly refer to the data in the database
        - Answer: Schema
    - Specifies a preset value if a value is omitted in the INSERT statement
        - Answer: DEFAULT
    - Stores data; basic unit of storage composed of rows and columns
        - Answer: Table
    - Command use to make a new table
        - Answer: CREATE
- *Try It / Solve It*
    - 1. Complete the GRADUATE CANDIDATE table instance chart. Credits is a foreign-key column referencing the requirements table

Column Name	student_id	last_name	first_name	credits	graduation_date
Key Type					
Nulls/Unique					
FK Column					
Datatype	NUMBER	VARCHAR2	VARCHAR2	NUMBER	DATE
Length	6			3	

        - Answer:

Column name	student_id	last_name	first_name	credits	graduation_date
Key Type	Primary Key			Foreign Key	
Nulls/Unique	NOT NULL	NOT NULL	NOT NULL		
FK Column				Yes	
Datatype	NUMBER	VARCHAR2	VARCHAR2	NUMBER	DATE
Length	6			3	

    - 2. Write the syntax to create the grad_candidates table

        - CREATE TABLE grad_candidates (
          student_id NUMBER(6) PRIMARY KEY,
          last_name VARCHAR2(50) NOT NULL,
          first_name VARCHAR2(50) NOT NULL,
          credits NUMBER(3) REFERENCES requirements(credits),
          graduation_date DATE
          );
    - 3. Confirm creation of the table using DESCRIBE
        - DESCRIBE grad_candidates;
    - 4. Create a new table using a subquery. Name the new table your last name -- e.g., smith_table. Using a subquery, copy grad_candidates into smith_table
        - CREATE TABLE smith_table AS
          SELECT * FROM grad_candidates;
    - 5. Insert your personal data into the table created in question 4
        - CREATE TABLE smith_table AS
          SELECT * FROM grad_candidates;
    - 6. Query the data dictionary for each of the following. In separate sentences, summarize what each query will return
        - USER_TABLES
            - The query will return a list of all the tables that are owned by the current user
        - USER_OBJECTS
            - The query will return information about all the objects owned by the current user
        - USER_CATALOG or USER_CAT
            - The query will return a summary listing information of objects in the current user's schema

Database Programming with SQL 13-2: Using Data Types Practice Activities

- *Vocabulary:* Identify the vocabulary word for each definition below
    - Allows time to be stored as an interval of years and months
        - Answer: Interval year to month
    - When a column is selected in a SQL statement the time is automatically converted to the user's timezone
        - Answer: Timestamp with local time zone
    - Binary large object data up to 4 gigabytes
        - Answer: Blob
    - Stores a time zone value as a displacement from Universal Coordinated Time or UCT
        - Answer: Timestamp with time zone

- Allows time to be stored as an interval of days to hours, minutes, and seconds
  - Answer: Interval day to second
- Character data up to 4 gigabytes
  - Answer: Clob
- Allows the time to be stored as a date with fractional seconds
  - Answer: Timestamp

- *Try It / Solve It*
  - 1. Create tables using each of the listed time-zone data types, use your time-zone and one other in your examples. Answers will vary
    - a. TIMESTAMP WITH LOCAL TIME ZONE
      - CREATE TABLE timezone_local_example (
        event_id NUMBER PRIMARY KEY,
        event_name VARCHAR2(50),
        event_timestamp TIMESTAMP WITH LOCAL TIME ZONE
        );
      - INSERT INTO timezone_local_example (event_id, event_name, event_timestamp)
        VALUES (1, 'Event in EST', TIMESTAMP '2024-11-10 10:00:00-05:00');
      - INSERT INTO timezone_local_example (event_id, event_name, event_timestamp)
        VALUES (2, 'Event in PST', TIMESTAMP '2024-11-10 10:00:00-08:00');
    - b. INTERVAL YEAR TO MONTH
      - CREATE TABLE membership_duration (
        member_id NUMBER PRIMARY KEY,
        member_name VARCHAR2(50),
        duration INTERVAL YEAR TO MONTH
        );
      - INSERT INTO membership_duration (member_id, member_name, duration)
        VALUES (1, 'Alice', INTERVAL '2-6' YEAR TO MONTH); -- 2 years, 6 months
      - INSERT INTO membership_duration (member_id, member_name, duration)
        VALUES (2, 'Bob', INTERVAL '1-3' YEAR TO MONTH); -- 1 year, 3 months
    - c. INTERVAL DAY TO SECOND
      - CREATE TABLE session_duration (
        session_id NUMBER PRIMARY KEY,

session_name VARCHAR2(50),
duration INTERVAL DAY TO SECOND
);

- INSERT INTO session_duration (session_id, session_name, duration)
  VALUES (1, 'Morning Session', INTERVAL '1 10:30:15' DAY TO SECOND); -- 1 day, 10 hours, 30 mins, 15 secs
- INSERT INTO session_duration (session_id, session_name, duration)
  VALUES (2, 'Afternoon Session', INTERVAL '0 04:20:10' DAY TO SECOND); -- 0 days, 4 hours, 20 mins, 10 secs

- 2. Execute a SELECT * from each table to verify your input
  - SELECT * FROM timezone_local;
  - SELECT * FROM membership_duration;
  - SELECT * FROM session_duration;
- 3. Give 3 examples of organizations and personal situations where it is important to know to which time zone a date-time value refers
  - Global teams: If an organization has employees from various parts of the world, they will be working in different time zones. For any scheduled events or deadlines, it will be very important to know the specific time zone to follow
  - International flights: For the itinerary of international flights, it is important to read the time zone details very carefully. The traveler will have to be careful when reading the times relating to the flight in order to make sure they are on time and not missing the flight
  - Webinar: If an international conference is being hosted, there will be attendees from various parts of the world. The attendees will have different local time zones, so it is important to specify which date and time zone the online conference will be taking place

Database Programming with SQL 13-3: Modifying a Table Practice Activities

- *Try It / Solve It*
  - Before beginning the practice exercises, execute a DESCRIBE for each of the following tables: o_employees, o_departments and o_jobs. These tables will be used in the exercises. If they do not exist in your account, create them as follows:
    - 1. Create the three o_tables – jobs, employees, and departments – using the syntax:
      - CREATE TABLE o_jobs AS (SELECT * FROM jobs);
      - CREATE TABLE o_employees AS (SELECT * FROM employees);

- CREATE TABLE o_departments AS (SELECT * FROM departments);
- 2. Add the Human Resources job to the jobs table:
  - INSERT INTO o_jobs (job_id, job_title, min_salary, max_salary) VALUES('HR_MAN', 'Human Resources Manager', 4500, 5500);
- 3. Add the three new employees to the employees table:
  - INSERT INTO o_employees (employee_id, first_name, last_name, email, hire_date, job_id)
  - VALUES(210, 'Ramon', 'Sanchez', 'RSANCHEZ', SYSDATE, 'HR_MAN');
- 4. Add Human Resources to the departments table:
  - INSERT INTO o_departments(department_id, department_name)
  - VALUES (210,'Human Resources');

- You will need to know which columns do not allow null values
  - 1. Why is it important to be able to modify a table?
    - It is important to have the ability to modify a table in order to have up-to-date information for business needs.
  - 2. CREATE a table called Artists
    - a. Add the following to the table: artist ID, first name, last name, band name, email, hourly rate
      - CREATE TABLE Artists (
        artist_id NUMBER PRIMARY KEY,
        first_name VARCHAR2(50),
        last_name VARCHAR2(50),
        band_name VARCHAR2(100),
        email VARCHAR2(100),
        hourly_rate NUMBER(5, 2)
        );
    - b. INSERT one artist from the d_songs table
      - INSERT INTO Artists (artist_id, first_name, last_name, band_name, email, hourly_rate)
        SELECT artist_id, first_name, last_name, band_name, email, hourly_rate
        FROM d_songs
        WHERE artist_id = 1;
    - c. INSERT one artist of your own choosing
      - INSERT INTO Artists (artist_id, first_name, last_name, band_name, email, hourly_rate)
        VALUES (2, 'John', 'Doe', 'The Wanderers', 'john.doe@example.com', 75.00);

- d. Give an example how each of the following may be used on the table that you have created:
    - 1) ALTER TABLE
        - ALTER TABLE Artists ADD (genre VARCHAR2(50));
    - 2) DROP TABLE
        - DROP TABLE Artists;
    - 3) RENAME TABLE
        - RENAME Artists TO Musicians;
    - 4) TRUNCATE
        - TRUNCATE TABLE Artists;
    - 5) COMMENT ON TABLE
        - COMMENT ON TABLE Artists IS 'storing information about artists;

- 3. In your o_employees table, enter a new column called “Termination.” The datatype for the new column should be VARCHAR2. Set the DEFAULT for this column as SYSDATE to appear as character data in the format: February 20th, 2003
    - ALTER TABLE o_employees
      ADD Termination VARCHAR2(20) DEFAULT
      TO_CHAR(SYSDATE, 'Month DDth, YYYY');

- 4. Create a new column in the o_employees table called start_date. Use the TIMESTAMP WITH LOCAL TIME ZONE as the datatype
    - ALTER TABLE o_employees
      ADD start_date TIMESTAMP WITH LOCAL TIME ZONE;
- 5. Truncate the o_jobs table. Then do a SELECT * statement. Are the columns still there? Is the data still there?
    - TRUNCATE TABLE o_jobs;
        - The columns will still be there but the data will not be there because TRUNCATE only deletes the data in the table and the structure of the table is not altered
- 6. What is the distinction between TRUNCATE, DELETE, and DROP for tables?
    - TRUNCATE removes all the rows from a table without having to perform individual row deletions
    - DELETE removes rows based on a condition or removes all the rows if no condition is specified
    - DROP completely deletes the table and all its data
- 7. List the changes that can and cannot be made to a column
    - For a column, you can change the data types, increase the size of columns, modify the default/initial values, add constraints, and more. However, you

cannot decrease the size of columns, change the data type to be incompatible, remove referenced constraints, rename column names to be duplicates, and more.

- 8. Add the following comment to the o_jobs table: "New job description added". View the data dictionary to view your comments
  - COMMENT ON TABLE o_jobs IS 'new job description added';
- 9. Rename the o_jobs table to o_job_description
  - ALTER TABLE o_jobs RENAME TO o_job_description;
- 10. F_staffs table exercises:
  - a. Create a copy of the f_staffs table called copy_f_staffs and use this copy table for the remaining labs in this lesson
    - CREATE TABLE copy_f_staffs AS
      SELECT *
      FROM f_staffs;
  - b. Describe the new table to make sure it exists
    - DESCRIBE copy_f_staffs;
  - c. Drop the table
    - DROP TABLE copy_f_staffs;
  - d. Try to select from the table [Returns an error that table does not exist]
    - SELECT *
      FROM copy_f_staffs;
  - e. Investigate your recycle bin to see where the table went
    - SELECT OBJECT_NAME, ORIGINAL_NAME
      FROM recyclebin
      WHERE ORIGINAL_NAME = 'COPY_F_STAFFS';
  - f. Try to select from the dropped table by using the value stored in the OBJECT_NAME column. You will need to copy and paste the name as it is exactly, and enclose the new name in “ “ (double quotes). So if the dropped name returned to you is BIN$Q+x1nJdcUnngQESYELVIdQ==$0, you need to write a query that refers to “BIN$Q+x1nJdcUnngQESYELVIdQ==$0”
    - SELECT *
      FROM "BIN$Q+x1nJdcUnngQESYELVIdQ==$0";
  - g. Un-drop the table
    - FLASHBACK TABLE copy_f_staffs TO BEFORE DROP;
  - h. Describe the table
    - DESCRIBE copy_f_staffs;
- 11. Still working with the copy_f_staffs table, perform an update on the table
  - a. Issue a select statement to see all rows and all columns from the copy_f_staffs table;

        - SELECT * FROM copy_f_staffs;
- b. Change the salary for Sue Doe to 12 and commit the change
    - UPDATE copy_f_staffs
      SET salary = 12
      WHERE first_name = 'Sue' AND last_name = 'Doe';
      COMMIT;
- c. Issue a select statement to see all rows and all columns from the copy_f_staffs table;
    - SELECT * FROM copy_f_staffs;
- d. For Sue Doe, update the salary to 2 and commit the change
    - UPDATE copy_f_staffs
      SET salary = 2
      WHERE first_name = 'Sue' AND last_name = 'Doe';
- e. Issue a select statement to see all rows and all columns from the copy_f_staffs table;
    - SELECT * FROM copy_f_staffs;
- f. Now, issue a FLASHBACK QUERY statement against the copy_f_staffs table, so you can see all the changes made
    - SELECT *
      FROM copy_f_staffs AS OF TIMESTAMP (SYSTIMESTAMP - INTERVAL '1' MINUTE);
- g. Investigate the result of f), and find the original salary and update the copy_f_staffs table salary column for Sue Doe back to her original salary
    - UPDATE copy_f_staffs
      SET salary = 10
      WHERE first_name = 'Sue' AND last_name = 'Doe';
      COMMIT;