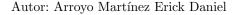


Universidad Nacional Autónoma de México Facultad de Ciencias

Fundamentos de Bases de Datos Práctica 2

Creación y Gestión de Esquemas Básicos





Introducción

En esta práctica, los estudiantes trabajarán en equipos de tres integrantes para crear y manipular esquemas de bases de datos en Oracle LiveSQL, incluyendo la definición de claves primarias y foráneas, y la manipulación avanzada de datos. Además de crear tablas relacionadas y demostrar la integridad referencial entre ellas, se abordará la modificación, eliminación y actualización de registros, así como el uso de consultas básicas con condiciones.

Objetivos

- 1. Crear y manipular esquemas básicos con claves primarias y foráneas.
- 2. Demostrar la integridad referencial y practicar la eliminación en cascada.
- 3. Realizar operaciones avanzadas de manipulación de datos usando INSERT, UPDATE, DELETE, ALTER, DROP, y CREATE.
- 4. Ejecutar consultas simples usando condiciones con WHERE.

Justificación

Debido a las restricciones de privilegios en Oracle LiveSQL, intentar crear usuarios o asignar permisos produce el error ORA-01031: insufficient privileges. Por esta razón, este aspecto será cubierto teóricamente en clase y no formará parte de esta práctica. En su lugar, nos enfocaremos en aspectos esenciales de la creación de esquemas y la integridad de las relaciones.

Especificaciones de Desarrollo

Paso 1: Crear Esquema con Tablas Relacionadas

1. En la sección Scratchpad de Oracle LiveSQL, crea una tabla llamada departments:

```
CREATE TABLE departments (
    department_id NUMBER PRIMARY KEY,
    department_name VARCHAR2(50) NOT NULL
);
```



2. Luego, crea una tabla llamada employees que tenga una clave foránea que referencie a departments:

```
CREATE TABLE employees (
    employee_id NUMBER PRIMARY KEY,
    first_name VARCHAR2(50),
    last_name VARCHAR2(50),
    department_id NUMBER,
    CONSTRAINT fk_department
        FOREIGN KEY (department_id)
        REFERENCES departments(department_id));
```

3. Crea una tabla llamada projects para gestionar los proyectos en los que trabajan los empleados:

```
Ejercicio 1
```

4. Crea una tabla que represente una relación de muchos a muchos entre employees y projects:

```
Ejercicio 2
```

Paso 2: Insertar Datos en las Tablas

1. Inserta algunos registros en la tabla departments:

```
INSERT INTO departments (department_id, department_name)
VALUES (1, 'Recursos Humanos');
INSERT INTO departments (department_id, department_name)
VALUES (2, 'Finanzas');
```

Ejercicio 3

Inserta el departamento 'IT'

2. Inserta algunos registros en la tabla employees:

```
INSERT INTO employees (employee_id, first_name, last_name, department_id)
VALUES (1, 'Juan', 'Pérez', 1);

INSERT INTO employees (employee_id, first_name, last_name, department_id)
VALUES (2, 'Ana', 'García', 2);

INSERT INTO employees (employee_id, first_name, last_name, department_id)
VALUES (3, 'Luis', 'Martínez', 3);
```



3. Inserta los proyectos Implementación de ERP, Rediseño del Sitio Web y Migración a la Nube.

```
Ejercicio 4

INSERT INTO projects (project_id, project_name)

VALUES (1, 'Implementación de ERP');

...
```

4. Asocia empleados con proyectos en la tabla employee_projects:

```
INSERT INTO employee_projects (employee_id, project_id)
VALUES (1, 1);

INSERT INTO employee_projects (employee_id, project_id)
VALUES (2, 2);

INSERT INTO employee_projects (employee_id, project_id)
VALUES (3, 3);

INSERT INTO employee_projects (employee_id, project_id)
VALUES (1, 2);
```

Paso 3: Manipulación de Datos

1. Actualiza los registros en la tabla employees para cambiar el departamento de Ana García a IT:

```
Ejercicio 5

UPDATE ...
```

2. Elimina el registro del proyecto Rediseño del Sitio Web y todos los registros relacionados en la tabla employee_projects:

```
Ejercicio 6

DELETE ...
```

3. Agrega una nueva columna email a la tabla employees para almacenar las direcciones de correo electrónico de los empleados:

```
Ejercicio 7

ALTER ...
```

4. Inserta los correos electrónicos correspondientes para cada empleado:

```
UPDATE employees SET email = 'juan.perez@empresa.com' WHERE employee_id = 1;
UPDATE employees SET email = 'ana.garcia@empresa.com' WHERE employee_id = 2;
UPDATE employees SET email = 'luis.martinez@empresa.com' WHERE employee_id = 3;
```



5. Elimina la tabla employee_projects y justifica por qué esto podría ser necesario en un contexto real:

Ejercicio 8

DROP ...

Paso 4: Consultas con Condiciones

1. Realiza una consulta para listar todos los empleados que trabajan en el departamento de IT:

Ejercicio 9

Select ...

2. Realiza una consulta para encontrar los empleados que no están asignados a ningún proyecto:

Ejercicio 10

Select ...

 $3.\,$ Realiza una consulta para listar todos los departamentos que tienen menos de 2 empleados:

Ejercicio 11
Select ...

Paso 4: Demostración de la Integridad Referencial

1. Intenta eliminar un registro de la tabla departments que esté referenciado en la tabla employees:

DELETE FROM departments WHERE department_id = 1;

- 2. Observa el error de consistencia generado y describe el comportamiento:
 - -- Deberías recibir un error similar a:
 - -- ORA-02292: integrity constraint (SCHEMA.FK_DEPARTMENT) violated child record found
- 3. Ahora, explica qué deberíamos hacer para poder eliminar todo un departamento. (+0.5 por explicación correcta/+1.0 con operaciones en sintaxis de SQL)

Entregables

- Capturas de pantalla que evidencien:
 - La creación y modificación de las tablas departments, employees, projects, y employee_projects.



- La inserción y manipulación de datos en las tablas.
- Las consultas realizadas y sus resultados.
- Los intentos de eliminación de registros y cualquier error de integridad referencial encontrado.
- Un archivo PDF con las capturas de pantalla y una explicación detallada de cada paso, incluyendo las justificaciones y conclusiones derivadas del trabajo en equipo.

Rúbrica de Evaluación

Criterio	Descripción	%
DDL	CREATE, ALTER y DROP	25%
DML	SELECT, INSERT, DELETE y UPDATE	25%
Consulta de datos	Consulta y visualización correcta de los datos	20%
Integridad referencial	Demostración de la integridad referencial y manejo de errores	20%
Presentación	Claridad y organización de las capturas de pantalla y explicaciones	10 %

Recursos

- Live SQL
- SQL Language Reference
- SQL Tutorial
- mockaroo