



Fundamentos de Bases de Datos  
**Práctica 3**  
Creación de Tablas Complejas y  
Consultas Avanzadas



Autor: Arroyo Martínez Erick Daniel

---

## Introducción

En esta práctica, los estudiantes trabajarán con un esquema de base de datos complejo previamente implementado. El objetivo es ejecutar y analizar una serie de consultas SQL que emplean técnicas avanzadas. Estas consultas abordarán una variedad de temas, desde la selección y filtrado de datos hasta la agregación y uso de funciones avanzadas.

## Objetivos

1. Implementar un esquema 1 de base de datos complejo con múltiples tablas y relaciones en Oracle LiveSQL.
2. Realizar consultas SQL avanzadas para extraer y analizar datos del esquema complejo.
3. Aplicar una variedad de técnicas SQL, incluyendo agregaciones, condiciones de filtrado, y manipulación de datos.
4. Comprender y utilizar correctamente funciones y operadores SQL avanzados.

## Especificaciones de Desarrollo

### Paso 1: Implementación del Esquema

1. Crea las siguientes tablas en Oracle LiveSQL según el diagrama relacional proporcionado:
  - **Departments**
  - **Employees**
  - **Projects**
  - **Project\_Assignments**
  - **Salaries**
2. Asegúrate de incluir las restricciones de integridad correspondientes para cada tabla, como NOT NULL, AUTO\_INCREMENT, UNIQUE, y CHECK.

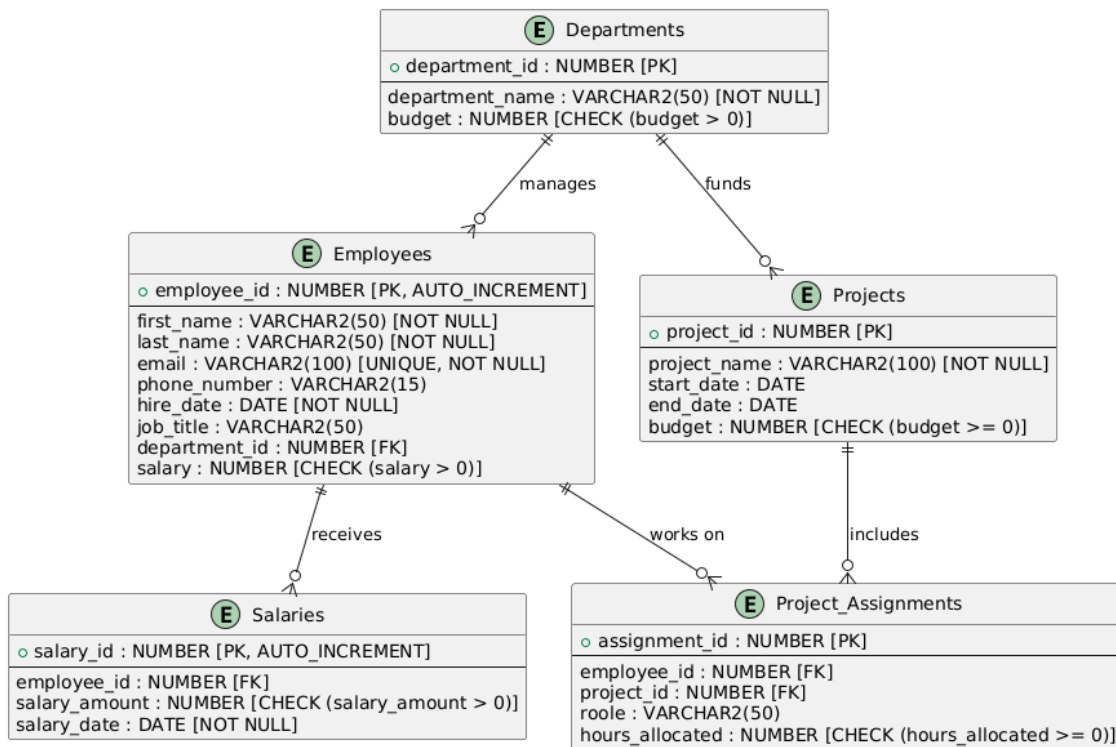


Figura 1: Diagrama Relacional

## Paso 2: Inserción de Datos

1. Inserta datos de ejemplo en cada tabla. Asegúrate de que los datos respeten las restricciones de integridad establecidas.
2. Los datos deben ser suficientes para poder realizar las consultas básicas requeridas posteriormente.

## Paso 3: Consultas Avanzadas

### Restricción

No deben usar Joins

1. Selecciona los nombres y apellidos de todos los empleados que trabajan en un departamento con un presupuesto mayor a 1,000,000.  
**Hint:** Usa una subconsulta para filtrar los departamentos con un presupuesto superior a un millón.
2. Obtén los nombres y apellidos de los empleados que tienen un salario mayor al promedio de todos los salarios. Ordena los resultados por el salario en orden descendente.  
**Hint:** Utiliza la función de agregación que calcula el promedio para comparar los salarios individuales con el promedio.
3. Encuentra todos los proyectos que tienen un presupuesto entre 500,000 y 1,000,000. Muestra el nombre del proyecto y su presupuesto.  
**Hint:** Usa la cláusula para filtrar los presupuestos dentro del rango especificado.
4. Inserta un nuevo empleado en la tabla **Employees** con un salario y asigna este empleado a un departamento específico.  
**Hint:** Asegúrate de asignar correctamente el ID del departamento en la nueva fila que insertes.

5. Actualiza el salario de todos los empleados cuyo título de trabajo es 'Manager' incrementándolo en un 10%.  
**Hint:** Usa una operación matemática para incrementar los salarios en función de su valor actual.
6. Elimina todos los registros de empleados que no tienen ningún proyecto asignado.  
**Hint:** Utiliza una subconsulta o un operador de comparación para identificar empleados sin asignaciones de proyectos.
7. Cuenta cuántos empleados trabajan en cada departamento y muestra el nombre del departamento junto con el número de empleados. Filtra los resultados para mostrar solo los departamentos con más de 10 empleados.  
**Hint:** Usa la función de agregación que cuenta filas para agrupar los empleados por departamento.
8. Encuentra el salario mínimo y máximo que reciben los empleados en el departamento de 'IT'. Muestra el título del trabajo, el salario mínimo y el salario máximo.  
**Hint:** Usa funciones de agregación para calcular los valores mínimo y máximo.
9. Encuentra los nombres de los proyectos que comenzaron en el año 2023 y que están en el departamento con el presupuesto más alto.  
**Hint:** Usa una subconsulta para obtener el presupuesto más alto y filtra los proyectos por la fecha de inicio.
10. Selecciona los nombres y apellidos de los empleados que tienen un correo electrónico que contiene el dominio 'example.com'. Asegúrate de que el correo electrónico sea único.  
**Hint:** Filtra los resultados usando un patrón que coincida con el dominio 'example.com' en la dirección de correo electrónico.
11. Muestra los nombres de los empleados que están trabajando en más de un proyecto. Ordena los resultados alfabéticamente.  
**Hint:** Agrupa los resultados por empleado y filtra aquellos que tienen más de una asignación de proyecto.
12. Selecciona el nombre del proyecto y el total de horas asignadas a cada proyecto. Filtra los resultados para mostrar solo los proyectos con más de 1000 horas asignadas.  
**Hint:** Usa una función de agregación para sumar las horas asignadas y filtra los proyectos que cumplan con el criterio.

## Justificación

En esta sección, deben explicar las relaciones entre las diferentes tablas del esquema y la cardinalidad de cada relación. Deben describir cómo estas relaciones y la cardinalidad asociada impactan en el diseño del esquema y la integridad de los datos. Además, deben explicar cómo las restricciones de integridad, como las claves foráneas y las verificaciones, garantizan la consistencia y coherencia de los datos en todo el sistema.

- **Departments y Employees:**
- **Departments y Projects:**
- **Employees y Project\_Assignments:**
- **Projects y Project\_Assignments:**
- **Employees y Salaries:**

## Entregables

- Un archivo `.sql` que contenga la implementación completa del esquema de base de datos, incluyendo la creación de tablas y la inserción de datos.
- Capturas de pantalla que evidencien:
  - La creación de cada tabla con sus restricciones.
  - Las consultas realizadas y los resultados obtenidos.
- Un archivo PDF con las capturas de pantalla y una breve explicación de cada paso realizado.

## Rúbrica de Evaluación

Criterio	Descripción	Ponderación
<b>Creación de tablas</b>	Implementación correcta de las tablas con restricciones de integridad	15 %
<b>Inserción de datos</b>	Inserción correcta y coherente de datos en las tablas	10 %
<b>Consultas básicas</b>	Realización correcta de las consultas y resultados obtenidos	50 %
<b>Justificación</b>	Claridad y corrección de argumentos	15 %
<b>Presentación</b>	Claridad y organización de las capturas de pantalla y explicaciones	10 %

## Recursos

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