



IIC2115 - Programación como Herramienta para la Ingeniería

Consultas en SQL

Profesor: Hans Löbel

Structured Query Language (SQL)

- Lenguaje de definición de datos (DDL)
 - Creación
 - Inserción
 - Eliminación
 - Modificación de definiciones de tablas.

*Las restricciones de integridad se pueden definir en tablas, ya sea cuando se crea la tabla o posteriormente.

- Lenguaje de manipulación de datos (DML)
 - Consultas



Consultas

SELECT [DISTINCT]

column_list

FROM

table_list

WHERE

row_filter

Name Ag	е
---------	---

Golith K	32
Rabin R	31

SELECT * FROM Empleados

SELECT * FROM Empleados E **WHERE** E.Age > 30

SELECT Name, Age **FROM** Empleados E **WHERE** E.Age > 30

Joins

SELECT [DISTINCT]

column_list

FROM

table_list

WHERE

row_filter

Emp No	Name	Age	Department id	Salary
001	Alex S	26	1	5000
				3000
002	Golith K	32	2	5600
003	Rabin R	31	2	5600
004	Jons	26	3	5100

SELECT Name **FROM** Empleados E, Departments D **WHERE** E.Department_id = D.id **AND** D.Department = 'Store'

Anidación

Emp No	Name	Age	Department	Salary
001 Alex S		26	Store	5000
002	Golith K	32	Marketing	5600
003	Rabin R 31 M		Marketing	5600
004 Jons		26	Security	5100

SELECT Name FROM (SELECT Name, Salary FROM Empleados E WHERE E.Age < 30)
WHERE Salary >= 5100

SELECT Name FROM Empleados E WHERE E.Salary >= 5100 AND E.Age < 30

Otras funciones importantes

ORDER BY

GROUP BY - HAVING

COUNT

SUM

AVG

MAX

MIN

Uso en Python: DML

```
connection = sqlite3.connect('ejemplo.db')
cursor = connection.cursor()
sqlStatement = 'SELECT * FROM Empleados'
cursor.execute(sqlStatement)
una_fila = cursor.fetchone()
todas_filas = cursor.fetchall()
connection.close()
```

Uso en Python: parametrización

```
def mayores_que(edad):
    connection = sqlite3.connect('ejemplo.db')
    cursor = connection.cursor()

sqlStatement = 'SELECT * FROM Empleados E WHERE E.Age > {}'.format(edad)

cursor.execute(sqlStatement)
    resp = cursor.fetchall()
    connection.close()
    return resp
```

Uso en Python: parametrización

```
def mayores_que(edad):
    connection = sqlite3.connect('ejemplo.db')
    cursor = connection.cursor()

sqlStatement = f'SELECT * FROM Empleados E WHERE E.Age > {edad}'

cursor.execute(sqlStatement)
    resp = cursor.fetchall()
    connection.close()
    return resp
```

Uso en Python: parametrización

```
def mayores_que(edad):
    connection = sqlite3.connect('ejemplo.db')
    cursor = connection.cursor()

sqlStatement = 'SELECT * FROM Empleados E WHERE E.Age > ?'

cursor.execute(sqlStatement, (edad,))
    resp = cursor.fetchall()
    connection.close()
    return resp
```

Manejo de errores

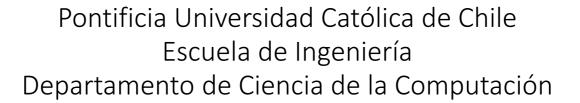
Al desarrollar este capítulo, se encontrarán dos tipos de errores:

- Errores de Python (de los que ya están familiarizados)
- Errores de la sintaxis de la base de datos (SQL)

CONSEJO: Pueden testear sus consultas directamente en la base de datos (p.ej., con DB Browser for SQLite) y luego utilizarla en Python.

Vamos a Colab...







IIC2115 - Programación como Herramienta para la Ingeniería

Consultas en SQL

Profesor: Hans Löbel