





¿Son Importantes las Consultas?

- Una cosa es guardar la información en forma estructurada y ordenada.
- Pero el utilizar la información y entender su relevancia son la clave de las Bases de Datos.
- De allí su máxima importancia y en ellas radica mucho de complejidad.
- Es la instrucción más utilizada y su comprensión es importantísima.



¿Cuál es el Resultado de SELECT?

- El resultado de un SELECT, una consulta es una tabla.
- ¿Puede haber un SELECT dentro de un SELECT?

SELECT es lo más difícil de BD

4

Sintaxis de SELECT

■ **SELECT** [ALL | DISTINCT]
lista_elementos_selección
FROM lista_referencias_tabla
[WHERE expresión_condicional]
[GROUP BY lista_columnas]
[HAVING expresión_condicional]
[ORDER BY lista_columnas]

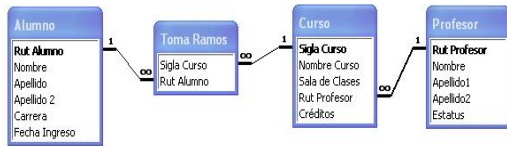
5

SELECT, FROM y WHERE

- Produce una tabla que se obtiene como producto cartesiano de las tablas especificadas en la cláusula FROM.
- Con la tabla anterior **elimina** todas las líneas que no satisfacen la expresión condicional de la cláusula WHERE (es decir las líneas por las cuales la expresión condicional no se cumple).

6

Ejemplo de Base de Datos



7

Ejemplo SELECT

- `SELECT *`
`FROM "Alumno"`
`WHERE "RutAlumno" >200`
- `SELECT *`
`FROM "TomaRamo", "Alumno"`
`WHERE "TomaRamo"."RutAlumno" =`
`"Alumno"."RutAlumno"`
- Toda la lista de las columnas de una tabla, los atributos, se pueden especificar usando el carácter `'*'`.

8

Ejemplo SELECT

- `SELECT "Alumno"."Nombre", "Alumno"."Apellido1", "Curso"."Nombre"`
`FROM "TomaRamo", "Curso", "Alumno"`
`WHERE "TomaRamo"."SiglaRamo" =`
`"Curso"."SiglaCurso" AND "TomaRamo"."RutAlumno" =`
`"Alumno"."RutAlumno"`
- Se pueden especificar atributos específicos y también se pueden hacer consultas entre varias tablas.

9

SELECT, GROUP BY

- GROUP BY (opcional) las líneas de la tabla resultante se reagrupan según los valores presentes en las columnas especificadas en la cláusula GROUP BY. Líneas con valores iguales se unen en una única línea.
- Se tiene que presentar en el SELECT una función por la cual agrupar

10

Ejemplo SELECT

- ```
SELECT "Carrera", COUNT("Carrera")
FROM "Alumno"
GROUP BY "Carrera"
```

11

---

---

---

---

---

---

---

---

## ***SELECT, HAVING***

- HAVING (opcional) Después de que GROUP BY combine los registros, HAVING muestra cualquier registro agrupado por la cláusula GROUP BY que satisfaga las condiciones de la cláusula HAVING

12

---

---

---

---

---

---

---

---

### Ejemplo *SELECT*

```
■ SELECT "Carrera", COUNT("Carrera")
 FROM "Alumno"
 GROUP BY "Carrera"
 HAVING COUNT("Carrera")>1;
```

13

---

---

---

---

---

---

---

### *SELECT, AS (Alias)*

■ En el caso en que se den ambigüedades, por la presencia de columnas con los mismos nombres en dos o más tablas incluidas en la cláusula FOR, se pueden resolver prefijando el nombre o el alias de la columna con el nombre o el alias de la tabla, separados por un punto. Por ejemplo, T.C indica la columna C de la tabla T. El alias de columna es el nombre que se le da a la columna.

14

---

---

---

---

---

---

---

### Ejemplo *SELECT*

```
■ SELECT "Carrera", COUNT("Carrera") AS "Conteo"
 FROM "Alumno"
 GROUP BY "Carrera"
 HAVING COUNT("Carrera")>1;

■ SELECT "Carrera", COUNT("Carrera") AS "Conteo"
 FROM "Alumno"
 GROUP BY "Carrera"
 HAVING "Conteo" >1;
```

{No funciona el último en OpenOffice}

15

---

---

---

---

---

---

---

## ***SELECT, DISTINCT o ALL***

- DISTINCT o ALL (opcional) se eliminan las líneas que resultan duplicadas. En el caso en que no estén presentes ni ALL ni DISTINCT, se asume ALL.

16

---

---

---

---

---

---

---

## ***Ejemplo SELECT***

- ```
SELECT ALL "Alumno"."RutAlumno",  
"Alumno"."Nombre", "Alumno"."Apellido1",  
"Profesor"."RutProfesor", "Profesor"."Apellido1"  
FROM "Profesor", "Curso", "Alumno", "TomaRamo"  
WHERE "Profesor"."RutProfesor" = "Curso"."RutProfesor"  
AND "Alumno"."RutAlumno" = "TomaRamo"."RutAlumno"  
AND "Curso"."SiglaCurso" = "TomaRamo"."SiglaRamo"
```

17

Ejemplo SELECT

- ```
SELECT DISTINCT "Alumno"."RutAlumno",
"Alumno"."Nombre", "Alumno"."Apellido1",
"Profesor"."RutProfesor", "Profesor"."Apellido1"
FROM "Profesor", "Curso", "Alumno", "TomaRamo"
WHERE "Profesor"."RutProfesor" = "Curso"."RutProfesor"
AND "Alumno"."RutAlumno" = "TomaRamo"."RutAlumno"
AND "Curso"."SiglaCurso" = "TomaRamo"."SiglaRamo"
```

18

---

---

---

---

---

---

---

## ***SELECT, ORDER BY***

- ORDER BY (opcional) las líneas de la tabla se ordenan según los valores presentes en las columnas especificadas en la cláusula. Con opciones de ascendentes (ASC), es por defecto, o descendientes (DESC).
- ORDER BY nombre\_columna [ ASC | DESC ] [ , nombre\_columna [ ASC | DESC ] ... ]

19

---

---

---

---

---

---

---

## ***Ejemplo SELECT***

- SELECT \* FROM "Alumno" ORDER BY "Alumno"."RutAlumno" ASC;
- SELECT \* FROM "Alumno" ORDER BY "Alumno"."RutAlumno" DESC;

20

---

---

---

---

---

---

---

## ***SELECT, LIKE, % y \_***

- Estas restricciones nos ayudan a filtrar los resultados en la tabla por los Nombres que empieza con el carácter "C". El operador LIKE permite una comparación entre cadenas de caracteres usando pattern contruidos con los caracteres "%" e "\_". El primero sustituye un número no precisado de caracteres (también 0), mientras que el segundo sustituye uno solo.

21

---

---

---

---

---

---

---

## Ejemplo *SELECT*

```
SELECT *
FROM "Alumno"
WHERE "Alumno"."Nombre" LIKE 'C%';
SELECT *
FROM "Alumno"
WHERE "Alumno"."Nombre" LIKE 'Camil_'
ORDER BY "Alumno"."RutAlumno" DESC;
```

22

---

---

---

---

---

---

---

## Ejemplo *SELECT, Access*

```
SELECT *
FROM Alumno
WHERE Alumno.Nombre LIKE "C*"
SELECT *
FROM Alumno
WHERE Alumno.Apellido LIKE "T????"
ORDER BY Alumno.Rut DESC;
```

23

---

---

---

---

---

---

---

## *SELECT, Funciones*

Las funciones de agregación previstas por el estándar SQL son COUNT, SUM, AVG, MAX y MIN, las cuales calculan respectivamente los números, la suma, la media aritmética, el máximo y el mínimo de los valores escalares presentes en la columna a la que se aplican.

24

---

---

---

---

---

---

---



## Ejemplo Dos *SELECT*

```
SELECT *
FROM "Alumno"
WHERE "Alumno"."Nombre" LIKE 'A*'
ORDER BY "Alumno"."RutAlumno" DESC;
```

25

---

---

---

---

---

---

---

## Ejemplo Tres *SELECT*

Consulta en que se pregunta el texto.

```
SELECT *
FROM "Alumno"
WHERE "Alumno"."Nombre" = ?;
```

26

---

---

---

---

---

---

---

**SQL, Consultas**  
**OpenOffice**

**Andrés Moreno S.**

27

---

---

---

---

---

---

---