SQL

COMIT: Ingeniería de información

Ingeniería de Sistemas y Computación
Universidad de Los Andes

SQL

Lenguajes:

- → **Cálculo Relacional**
- → **Álgebra Relacional**
- → SQL

Concepto: Lenguaje estructurado de consulta.

Estándares SQL86, SQL89,SQL92, SQL1999 y SQL2003.

SQL

Componentes:

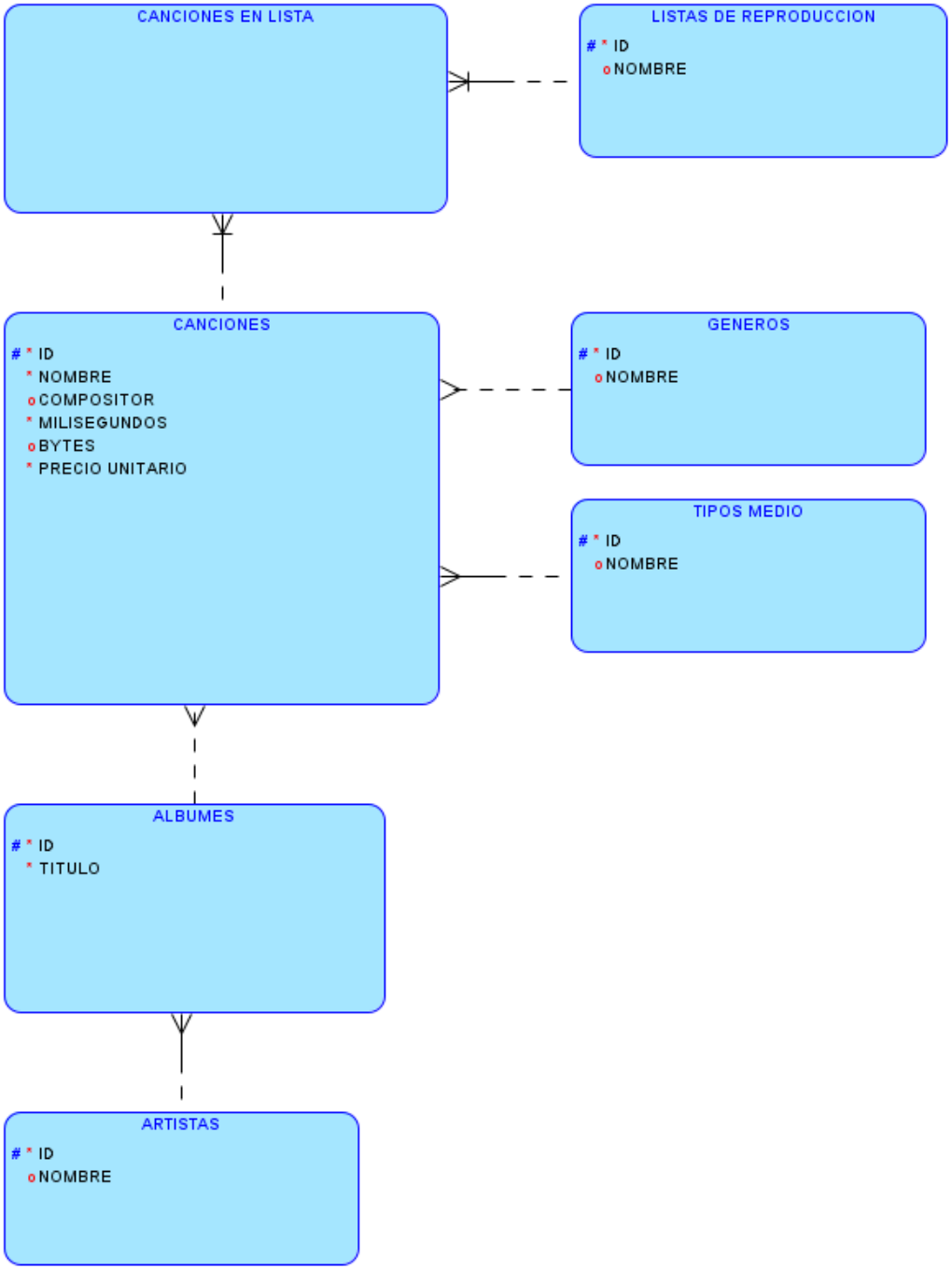
- DDL
  - Esquemas
  - Reglas de integridad
  - Vistas
  - Acceso a relaciones, vistas, etc.
- DML
- Control de transacciones
- SQL embebido y SQL dinámico.

Restricciones

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK

http://docs.oracle.com/cd/B19306_01/server.102/b14200/clauses002.htm

Modelo Entidad Relación MusiK



Modelo relacional MusiK

Canciones								
Id	**Nombre**	**Compositor**	**Milisegundos**	**Bytes**	**Precio_Unitario**	**Genero_Id**	**Medio_Id**	**Album_Id**
PK	NN		NN		NN		NN	NN

Canciones_en_lista	
Cancion_Id	**Lista_Id**
PK, FK (canciones.Id)	PK, FK (Lista_Reproduccion.Id)

Listas_de_reproduccion	
Id	**Nombre**
PK	NN

Generos	
Id	**Nombre**
PK	NN

Artistas	
Id	**Nombre**
PK	NN

Tipos_Medio	
Id	**Nombre**
PK	NN

Albumes		
Id	**Título**	**Artista_Id**
PK	NN	FK (Artistas.Id), NN

SQL: Creación de la BD

- Crear tablas
- Definir restricciones
- Crear secuencias
- Crear índices

Crear tablas - Canciones

```
CREATE TABLE CANCIONES
(	ID NUMBER NOT NULL,
	NOMBRE VARCHAR2(200 BYTE) NOT NULL,
	ALBUM_ID NUMBER,
	MEDIO_ID NUMBER NOT NULL,
	GENERO_ID NUMBER,
	COMPOSITOR VARCHAR2(220 BYTE),
	MILISEGUNDOS NUMBER NOT NULL,
	BYTES NUMBER,
	PRECIO_UNITARIO NUMBER(10,2) NOT NULL,
	CONSTRAINT PK_Cancion PRIMARY KEY (ID)
);
```

Definición de restricciones - Canciones

```
ALTER TABLE CANCIONES
ADD CONSTRAINT FK_Album_Id
FOREIGN KEY (ALBUM_ID) REFERENCES ALBUMES (ID);

ALTER TABLE CANCIONES
ADD CONSTRAINT FK_Genero_Id
FOREIGN KEY (GENERO_ID) REFERENCES GENEROS (ID);

ALTER TABLE CANCIONES
ADD CONSTRAINT FK_Medio_Id
FOREIGN KEY (MEDIO_ID) REFERENCES TIPOS_MEDIO (ID);
```

Crear tablas - Albumes

```
CREATE TABLE ALBUMES
(		ID NUMBER NOT NULL,
		TITULO VARCHAR2(160 BYTE) NOT NULL,
		ARTISTA_ID NUMBER NOT NULL,
		CONSTRAINT FK_ALBUMARTISTAID FOREIGN KEY
(ARTISTA_ID) REFERENCES ARTISTAS (ID) ENABLE
);
ALTER TABLE ALBUMES ADD CONSTRAINT PK_Album PRIMARY KEY
(ID);
```

Crear tablas - Artistas

```
CREATE TABLE ARTISTAS
(		ID NUMBER NOT NULL,
		NOMBRE VARCHAR2(120 BYTE)
);
ALTER TABLE ARTISTAS ADD CONSTRAINT PK_Artista PRIMARY KEY
(ID);
```

Crear tablas - Canciones_en_lista

```
CREATE TABLE CANCIONES_EN_LISTA
(		LISTA_ID NUMBER NOT NULL,
		CANCION_ID NUMBER NOT NULL,
		CONSTRAINT FK_CANCION_ID FOREIGN KEY (CANCION_ID)
		REFERENCES CANCIONES (ID),
		CONSTRAINT FK_LISTA_ID FOREIGN KEY (LISTA_ID)
		REFERENCES LISTAS_DE_REPRODUCCION (ID)
) ;
ALTER TABLE CANCIONES_EN_LISTA ADD CONSTRAINT
PK_Canciones_en_lista PRIMARY KEY (LISTA_ID, CANCION_ID);
```

Crear tablas - Generos

```
CREATE TABLE GENEROS
(           ID NUMBER NOT NULL,
            NOMBRE VARCHAR2(120 BYTE)
);
ALTER TABLE GENEROS ADD CONSTRAINT PK_Genero PRIMARY KEY
(ID);
```

Crear tablas – Listas_de_reproduccion

```
CREATE TABLE LISTAS_DE_REPRODUCCION
(		ID NUMBER NOT NULL,
		NOMBRE VARCHAR2(120 BYTE)
);
ALTER TABLE LISTAS_DE_REPRODUCCION ADD CONSTRAINT PK_Lista
PRIMARY KEY (ID);
```

Crear tablas – Tipos_medio

```
CREATE TABLE TIPOS_MEDIO
(           ID NUMBER NOT NULL,
            NOMBRE VARCHAR2(120 BYTE)
);
ALTER TABLE TIPOS_MEDIO ADD CONSTRAINT PK_Tipos_medio
PRIMARY KEY (ID);
```

Borrar tablas

```
DROP TABLE Artistas;
```

```
DROP TABLE Artistas CASCADE CONSTRAINTS;
```

SQL: Modificación de la BD

- Inserción
- Actualización
- Borrado

SQL: Modificación de la BD - Inserción

INSERT INTO R [(A1,…, An)]
VALUES (v1,…, vn)

INSERT INTO R S

Añadir a la base de datos la canción Pigs compuesta por Roger Waters

INSERT INTO canciones **(**id, nombre, album_id, medio_id, genero_id, compositor, milisegundos, bytes, precio_unitario**)**
VALUES (5, 'Pigs', 1, 4, 5, 'Waters', 435908, 5242880, '1,20'**)**

SQL: Modificación de la BD - Actualización

UPDATE R

SET A1 = V1 [,… An = Vn]

[WHERE Predicado]

Sume un 1 dólar al precio unitario de todas las canciones que tienen un precio unitario entre 1.20 y 2 dolares

```
UPDATE canciones
SET precio_unitario = precio_unitario + 1
WHERE precio_unitario BETWEEN 1,20 AND 2
```

SQL: Modificación de la BD - Borrado

DELETE FROM R
[WHERE Predicado]

Borrar las canciones compuestas por ‘David Gilmour’

DELETE FROM canciones
WHERE compositor like ‘%Gilmour%’;

Select

```
SELECT      <atributos>
FROM        <Relación>

[WHERE      <Condición>]
[ORDER BY   <atributos>]
[GROUP BY   <atributos>
        [HAVING     <Condición> ]]
```

Funciones de cadenas de caracteres

- http://www.tutorialspoint.com/sql/sql-string-functions.htm
- http://www.oracletutorial.com/oracle-string-functions/

Funciones de fechas

- http://www.tutorialspoint.com/sql/sql-date-functions.htm
- http://www.oracletutorial.com/oracle-date-functions/

SQL: consultas simples

- Se quiere conocer la lista de todos los artistas ordenados por su nombre

```
SELECT * FROM artistas ORDER BY nombre
```

- Se quiere obtener el nombre y el precio unitario de todas las canciones compuestas por Roger Waters.

```
SELECT nombre, precio_unitario FROM canciones
WHERE compositor LIKE '%Waters%';
```

Ejercicios: Se quiere saber…

1. Toda la información de todos los álbumes
2. El nombre de todos los géneros
3. El nombre de todos los tipos de medios
4. El nombre, el compositor y la duración de todas las canciones
5. El nombre de las canciones que no tienen registrado su peso en bytes

SQL: Reunión de relaciones - JOIN

- **Join: Selección aplicada al producto cartesiano de dos tablas**
- **Reunión interna**
  - Reunión interna
  - Reunión natural interna

SQL: Reunión interna

Se quiere conocer el nombre de todas las canciones que pertenecen la genero ‘Metal’

```
SELECT generos.nombre, canciones.nombre
FROM canciones, generos
WHERE generos.nombre like 'Metal' AND
canciones.genero_id=generos.id;
```

SQL: Reunión interna

Se quiere conocer el nombre de todas las listas de reproducción junto con el nombre de las canciones que contienen. El resultado debe estar ordenado por el nombre de la lista de reproducción de manera ascendente.

SQL: Reunión interna

Se quiere conocer el nombre de todas las listas de reproducción junto con el nombre de las canciones que contienen. El resultado debe estar ordenado por el nombre de la lista de reproducción de manera ascendente.

```
SELECT L.nombre as Nombre_Lista, C.nombre as Nombre_Cancion
FROM canciones C, canciones_en_lista CL, listas_de_reproduccion L
WHERE C.id=CL.cancion_id AND CL.lista_id=L.id
ORDER BY L.nombre ASC;
```

SQL: Reunión natural interna

Se quiere conocer todas las canciones de los álbumes producidos por la banda de rock progresivo Pink Floyd. El resultado debe contener el nombre del artista, el nombre del álbum, el nombre de las canciones y del compositor, y estar ordenado por el nombre de la canción.

SQL: Reunión natural interna

Se quiere conocer todas las canciones de los álbumes producidos por la banda de rock progresivo Pink Floyd. El resultado debe contener el nombre del artista, el nombre del álbum, el nombre de las canciones y del compositor, y estar ordenado por el nombre de la canción.

```
SELECT AR.nombre as Artista, AB.titulo as Album, C.nombre as Cancion,
C.compositor as Compositor
FROM artistas AR, albumes AB, canciones C
WHERE AB.artista_id=AR.id AND AB.id=C.album_id AND AR.nombre='Pink
Floyd'
ORDER BY C.nombre;
```

SQL – Ejercicios – Se quiere conocer…

Canciones								
Id	**Nombre**	**Compositor**	**Milisegundos**	**Bytes**	**Precio_Unitario**	**Genero_Id**	**Medio_Id**	**Album_Id**
PK	NN		NN		NN		NN	NN

Canciones_en_lista	
Cancion_Id	**Lista_Id**
PK, FK (canciones.Id)	PK, FK (Lista_Reproduccion.Id)

Listas_de_reproduccion	
Id	**Nombre**
PK	NN

Generos	
Id	**Nombre**
PK	NN

Artistas	
Id	**Nombre**
PK	NN

Tipos_Medio	
Id	**Nombre**
PK	NN

Albumes		
Id	**Título**	**Artista_Id**
PK	NN	FK (Artistas.Id), NN

1. El nombre de todos los álbumes del artista ‘Caetano Veloso’
2. El nombre de todas las listas de reproducción que tienen canciones de ‘Rock’.
3. El nombre de todos los álbumes que contienen canciones con un tipo de medio ‘MPEG audio file’.
4. El nombre de los artistas que tocan canciones de genero ‘Latin’

Funciones de agregación

- **Funciones de agregación**
  - [**AVG | SUM**] **(** expresión | [**DISTINCT**] columna **)**
  - [**MIN | MAX**] **(** expresión **)**
  - **COUNT (** [**DISTINCT**] columna | * **)**

✓No pueden utilizarse en la cláusula WHERE

✓Aplican sobre un conjunto, no sobre una tupla…

Ejercicios – Se quiere saber …

1. Cuántos álbumes hay (use count)
2. El promedio de la duración de todas las canciones (avg count)
3. Qué capacidad debería tener el disco duro en bytes para albergar todas las canciones de la base de datos (use sum)
4. Cuál es el nombre y precio de la canción más cara y el nombre y precio de la más barata (use max y min)
5. Cuál es el nombre de la canción con tiempo de duración mas corto (use min)

Grupos

- Un grupo está formado de tuplas que tienen el mismo valor para una columna específica
  - Cláusula GROUP BY
- Cada grupo se trata como un subconjunto de la respuesta,
  - Sobre el cual pueden hacerse operaciones
  - Aplican las funciones de agregación, que se evalúan para cada grupo
- Pueden seleccionarse grupos en la respuesta
  - Cláusula HAVING

Ejercicios – Se quiere saber …

1. El promedio del precio unitario por genero (use avg y agrupe por genero_id)
2. El número de canciones que tiene cada compositor. El resultado debe estar ordenado descendentemente por el número de canciones compuestas y ascendentemente el nombre del compositor (use count y agrupe por compositor)
3. El compositor con más canciones (use count, no tenga en cuenta las canciones con compositor null WHERE compositor is not null, agrupe por compositor, ordene descendentemente y limite la lista al primer resultado FETCH FIRST 1 ROWS ONLY)

Ejercicios – Se quiere saber …

1. El nombre de los compositores de rock que tienen 6 o más canciones
2. Los géneros que tienen más de 50 canciones. Debe aparecer el nombre del género y el número de canciones que tiene
3. El top 3 artistas que tienen más álbumes. Debe aparecer el nombre del artista y el nombre de los albumes
4. El top 5 de canciones que aparecen en más listas de reproducción. Debe aparecer el nombre de las canciones y el nombre de las listas.

FIN DE LA PRESENTACIÓN