

Desarrollo de Software

Práctica 3.6 – Algebra relacional y uso de join básico

Jesús Alberto Aréchiga Carrillo

22310439 4N

Profesor

José Luis García Cerpas

Mayo 2024

Guadalajara, Jalisco

Introducción

En esta práctica se va a hacer uso de join en una base de datos para manipular cómo se puede mostrar la información en conjunto, también se va a utilizar el comando explain para explicar cada resultado que salga.

Desarrollo

Primero se tiene que crear la base de datos con sus respectivas entradas y datos:

```
create database torneo;

use torneo;

create table grupos(
  idGrupo int ,
  nombre varchar(50),
  primary key(idGrupo)
);

create table sedes(
  idSede int,
  nombre varchar(60),
  estado varchar(60),
  primary key(idSede)
);

create table equipos(
  idEquipo int,
  nombre varchar(60),
  logo varchar(60),
  idSede int,
  idGrupo int,
  primary key(idEquipo),
  foreign key(idSede) references sedes(idSede),
  foreign key(idGrupo) references grupos(idGrupo)
);

create table jugadores(
  idJugador int,
  idEquipo int,
  nombre varchar(60),
  numero int,
  goles int,
  primary key(idJugador),
  foreign key(idEquipo) references equipos(idEquipo)
);
```

```

create table fechas(
    idFecha int,
    fecha date,
    primary key(idFecha)
);

create table partido(
    idPartido int,
    idFecha int,
    idEquipoLocal int,
    idEquipoVisitante int,
    marcadorLocal int,
    marcadorVisitante int,
    hora time,
    idSede int,
    primary key(idPartido),
    foreign key(idFecha) references fechas(idFecha),
    foreign key(idEquipoLocal) references equipos(idEquipo),
    foreign key(idEquipoVisitante) references equipos(idEquipo),
    foreign key(idSede) references sedes(idSede)
);

insert into grupos (idGrupo, nombre) values
    (1,'1'),
    (2,'2'),
    (3,'3'),
    (4,'4');

insert into sedes (idSede, nombre, estado)
    values (1, 'Jalisco', 'Jalisco'),
           (2, 'Omnilife', 'Jalisco'),
           (3, 'America', 'CDMX');

insert into equipos (idEquipo, nombre, logo, idSede, idGrupo) values
    (1, 'Atlas', 'atlas.jpg', 1,1),
    (2, 'Chivas', 'chivas.jpg', 2,2),
    (3, 'America', 'america.jpg', 3,3)
;

insert into jugadores (idJugador, idEquipo, nombre, numero, goles) values
    (1,1,"Jugador Atlas 1", 1,1),
    (2,2,"Jugador Chivas 1", 2,2),
    (3,1,"Jugador Atlas 2", 1,1),
    (4,2,"Jugador Chivas 2", 2,0),
    (5,1,"Jugador Atlas 3", 1,0)
;

```

Después se comienza a hacer uso de los comandos, primero se muestra el ejemplo del comando y luego se hace el ejercicio con el uso de join:

```
Select equipos.nombre, sedes.nombre, grupos.nombre
```

```
from equipos, sedes, grupos
```

```
where equipos.idSede=sedes.idSede
```

```
and equipos.idGrupo=grupos.idGrupo;
```

```
MariaDB [torneo]> SELECT equipos.nombre , sedes.nombre, grupos.nombre
-> FROM equipos
-> JOIN sedes ON equipos.idSede = sedes.idSede
-> JOIN grupos ON equipos.idGrupo = grupos.idGrupo;
```

nombre	nombre	nombre
Atlas	Jalisco	1
Chivas	Omnilife	2
America	America	3

3 rows in set (0.001 sec)

Ahora seguiría utilizar el comando EXPLAIN para que muestre la tabla de qué es lo que está haciendo:

```
MariaDB [torneo]> EXPLAIN SELECT equipos.nombre, sedes.nombre, grupos.nombre
-> FROM equipos
-> JOIN sedes ON equipos.idSede = sedes.idSede
-> JOIN grupos ON equipos.idGrupo = grupos.idGrupo;
```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	equipos	ALL	idSede,idGrupo	NULL	NULL	NULL	3	Using where
1	SIMPLE	sedes	ALL	PRIMARY	NULL	NULL	NULL	3	Using where; Using join buffer (Flat, BNL join)
1	SIMPLE	grupos	eq_ref	PRIMARY	PRIMARY	4	torneo.equipos.idGrupo	1	

3 rows in set (0.000 sec)

Siendo:

id: El identificador de la consulta.

select_type: El tipo de selección (SIMPLE, en este caso, significa una consulta básica sin subconsultas).

table: Las tablas involucradas en la consulta.

type: El tipo de unión (ALL, eq_ref).

possible_keys: Las claves que podrían usarse.

key: La clave que realmente se usó.

key_len: La longitud de la clave usada.

ref: Las columnas que se comparan con la clave.

rows: El número de filas que se espera sean examinadas.

Extra: Información adicional sobre cómo se ejecuta la consulta.

Siguiente ejemplo:

```
select jugadores.nombre, jugadores.numero, equipos.nombre  
from jugadores, equipos  
where jugadores.idEquipo=equipos.idEquipo;
```

```
MariaDB [torneo]> SELECT jugadores.nombre, jugadores.numero, equipos.nombre  
-> FROM jugadores  
-> JOIN equipos ON jugadores.idEquipo = equipos.idEquipo;  
+-----+-----+-----+  
| nombre | numero | nombre |  
+-----+-----+-----+  
| Jugador Atlas 1 | 1 | Atlas |  
| Jugador Chivas 1 | 2 | Chivas |  
| Jugador Atlas 2 | 1 | Atlas |  
| Jugador Chivas 2 | 2 | Chivas |  
| Jugador Atlas 3 | 1 | Atlas |  
+-----+-----+-----+  
5 rows in set (0.000 sec)
```

Con su explicación:

```
MariaDB [torneo]> EXPLAIN SELECT jugadores.nombre, jugadores.numero, equipos.nombre  
-> FROM jugadores  
-> JOIN equipos ON jugadores.idEquipo = equipos.idEquipo;  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
| id | select_type | table | type | possible_keys | key | key_len | ref | rows | Extra |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
| 1 | SIMPLE | equipos | ALL | PRIMARY | NULL | NULL | NULL | 3 | |  
| 1 | SIMPLE | jugadores | ALL | idEquipo | NULL | NULL | NULL | 5 | Using where; Using join buffer (flat, BNL join) |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
2 rows in set (0.000 sec)
```

Conclusiones

El álgebra relacional es muy útil para hacer búsquedas de datos que tengan relaciones en diferentes tablas, haciendo que como resultado de una tabla que tiene entradas de diferentes tablas y poder visualizar toda la información necesaria en un solo lugar con una sola búsqueda.

También es posible hacer uso de otros comandos de JOIN y cada comando tiene su propio funcionamiento, como se vio en la parte teórica del desarrollo.