

Base de datos

Considerando el esquema de la siguiente base de datos:

Platos(plato:char,precio:integer)

Menús(menu:char,plato:char)

Gourmet(persona:char, plato:char)

PK(plato)

PK(menu,plato)

FK(plato)/Platos

PK(persona,plato)

FK(plato)/Platos

resuelva las siguientes consultas en álgebra relacional y SQL

q1: Platos y precios

q2: Platos y precios excepto del plato 'Gambas'

q3: Precio de los platos que forman cada menú

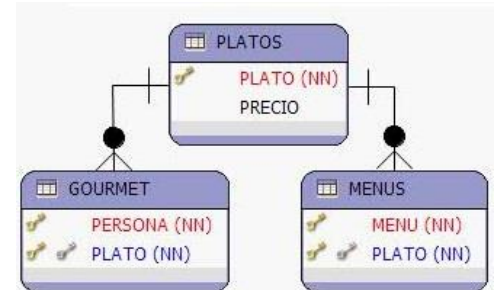
q4: Precio de los menús (suma de los precios de los platos)

q5: Precio de los menús (suma de los precios de los platos) que contienen más de dos platos

q6: Platos que gustan a todos

q7: Menús que gustan a Lola (le gustan todos sus platos)

q8: Menú más caro



| PLATOS (Table) created: 29 | | GOURMET (Table) created: | | MENUS (Table) created: 03 | |
|----------------------------|---------|--------------------------|-------------|---------------------------|-------------|
| Columns | Indexes | Columns | Indexes | Columns | Indexes |
| Where: | | | | | |
| PLATO | PRECIO | PERSONA | PLATO | MENU | PLATO |
| Entrecot | 20 | Lola | Sopa | M1 | Entrecot |
| Lubina | 18 | Lola | Setas | M2 | Sopa |
| Sopa | 3 | Lola | Helado | M2 | Lubina |
| Flan | 5 | Lola | Caviar | M3 | Flan |
| Helado | 4 | Lola | Ensaladilla | M3 | Ensaladilla |
| Ensaladilla | 6 | Luis | Gambas | M4 | Setas |
| Setas | 10 | Luis | Lubina | M4 | Gambas |
| Gambas | 25 | Luis | Flan | | |
| Caviar | 50 | Luis | Helado | | |
| | | Manolo | Setas | | |
| | | Manolo | Helado | | |
| | | Manolo | Flan | | |
| | | Manolo | Gambas | | |

Solución Álgebra Relacional

q1: Platos y precios $\prod_{plato, precio} Platos;$

q2: Platos y precios excepto del plato 'gambas' $\prod_{plato, precio} \left(\sigma_{plato \neq 'gambas'} (Platos) \right);$

q3: Precio de los platos que forman cada menú $\prod_{menu, plato, precio} (Menus \bowtie Platos);$

q4: Precio de los menús $\prod_{menu, sum(precio)} \left(\prod_{menu} (Menus \bowtie Platos) \right)$

q5: Precio de los menús que tienen más de dos platos $\prod_{menu, sum(precio)} \left(\sigma_{count(plato) > 2} \left(\prod_{menu} (Menus \bowtie Platos) \right) \right)$

q6: Platos que gustan a todos $\frac{Gourmet}{\prod_{persona} (Gourmet)} \equiv \prod_{plato} Platos - \left(\prod_{plato} \left(\left(\prod_{plato} Platos \right) \times \prod_{persona} Gourmet \right) - Gourmet \right)$

q7: Menús que gustan a Lola (le gustan todos sus platos)

a) Platos que le gustan a Lola $\prod_{plato} \left(\sigma_{persona = 'Lola'} (Gourmet) \right);$

b) Platos que no le gustan a Lola $\prod_{plato} (Platos) - \prod_{plato} \left(\sigma_{persona \neq 'Lola'} (Gourmet) \right);$

c) Menús que no gustan a Lola $\prod_{menu} \left(\left(\prod_{plato} (Platos) - \prod_{plato} \left(\sigma_{persona \neq 'Lola'} (Gourmet) \right) \right) \bowtie Menus \right);$

d) Menús que gustan a Lola $\prod_{menu} (Menus) - \prod_{menu} \left(\left(\prod_{plato} (Platos) - \prod_{plato} \left(\sigma_{persona \neq 'Lola'} (Gourmet) \right) \right) \bowtie Menus \right);$

q8: Menú más caro

$$\prod_{menu} \left(\sigma_{\max(sum(precio)) \leq sum(precio)} \left(\prod_{menu} (Menus \bowtie Platos) \right) \right)$$

Solución SQL Oracle

```
drop table gourmet cascade constraints;
drop table menus cascade constraints;
drop table platos cascade constraints;
create table platos (plato varchar2(20), precio integer, primary key(plato));
create table menus (menu varchar2(20), plato varchar2(20),
    primary key(menu, plato), foreign key(plato) references platos );
create table gourmet (persona varchar2(20), plato varchar2(20),
    primary key(persona, plato), foreign key(plato) references platos );

--
-- Inserta datos de ejemplo en las tablas
--
INSERT INTO platos VALUES ( 'Entrecot',20);
INSERT INTO platos VALUES ( 'Lubina',18);
INSERT INTO platos VALUES ( 'Sopa',3);
INSERT INTO platos VALUES ( 'Flan',5);
INSERT INTO platos VALUES ( 'Helado',4);
INSERT INTO platos VALUES ( 'Ensaladilla',6);
INSERT INTO platos VALUES ( 'Setas',10);
INSERT INTO platos VALUES ( 'Gambas',25);
INSERT INTO platos VALUES ( 'Caviar',50);

INSERT INTO menus VALUES ( 'M1', 'Sopa');
INSERT INTO menus VALUES ( 'M1', 'Entrecot');
INSERT INTO menus VALUES ( 'M2', 'Sopa');
INSERT INTO menus VALUES ( 'M2', 'Lubina');
INSERT INTO menus VALUES ( 'M2', 'Flan');
INSERT INTO menus VALUES ( 'M3', 'Ensaladilla');
INSERT INTO menus VALUES ( 'M3', 'Caviar');
INSERT INTO menus VALUES ( 'M4', 'Setas');
INSERT INTO menus VALUES ( 'M4', 'Gambas');

INSERT INTO gourmet VALUES ( 'Lola', 'Sopa');
INSERT INTO gourmet VALUES ( 'Lola', 'Setas');
INSERT INTO gourmet VALUES ( 'Lola', 'Helado');
INSERT INTO gourmet VALUES ( 'Lola', 'Caviar');
INSERT INTO gourmet VALUES ( 'Lola', 'Ensaladilla');
INSERT INTO gourmet VALUES ( 'Lola', 'Gambas');
INSERT INTO gourmet VALUES ( 'Luis', 'Gambas');
INSERT INTO gourmet VALUES ( 'Luis', 'Lubina');
INSERT INTO gourmet VALUES ( 'Luis', 'Flan');

INSERT INTO gourmet VALUES ( 'Luis', 'Helado');
INSERT INTO gourmet VALUES ( 'Manolo', 'Setas');
INSERT INTO gourmet VALUES ( 'Manolo', 'Helado');
INSERT INTO gourmet VALUES ( 'Manolo', 'Flan');
INSERT INTO gourmet VALUES ( 'Manolo', 'Gambas');

--Platos y precios
select plato, precio from platos;

--Platos y precios excepto del plato 'Gambas'
select plato, precio from platos
where plato <> 'Gambas';

--Precio de los platos que forman cada menú
select menu, plato, precio from menus natural join platos;
```

```
--Precio de los menús (suma de los precios de los platos)
select menu, sum(precio) from menus natural join platos group by menu;

--Precio de los menús (suma de los precios de los platos)
--que contienen más de dos platos
select menu, sum(precio) from menus natural join platos
group by menu having count(*) >2;

--Platos que gustan a todos
select P.plato from platos P where
    not exists (select G1.persona from gourmet G1 where
        not exists (select * from gourmet G2
            where G2.persona=G1.persona
            and G2.plato=P.plato));

--Menús que gustan a Lola
select distinct M1.menu from menus M1
where not exists (select M2.plato from platos P, menus M2
    where P.plato=M2.plato and M2.menu=M1.menu
    and not exists (select G.plato from gourmet G
        where G.persona='Lola' and G.plato=M2.plato));

select menu from menus
minus
(select distinct menu from menus natural join (select plato from platos
    minus select plato from gourmet where persona='Lola'))

--Menú más caro
select menu from menus natural join platos
group by menu having sum(precio) >= All (select sum(precio)
    from menus natural join platos group by menu);
select menu from menus natural join platos
group by menu
having sum(precio) = (select max(tot) from (select sum(precio) AS tot
    from menus natural join platos
    group by menu )
    );
```