

UF2. SQL
PROYECTO FINAL



MasterChef

Professor:
David Rodríguez Pellicer

Nombres:
Anthony Pozo
Dani Luque

1.1 CREACION DE TABLAS	3
1.2 SECUENCIAS	4
1.3 INSTERTS	5
CHEFS	5
RECETAS	5
INGREDIENTES	6
USUARIOS	7
EVALUACION	7
AUTORIA	8
CONTIENE	9
1.4 CONSULTAS	10
2. APARTADOS EXTRAS	20
2.1 Creación de nueva tabla	20
2.2 Datos adicionales	21
CHEFS	21
RECETAS	21
INGREDIENTES	21
USUARIOS	21
EVALUACIÓN	21
AUTORÍA	21
CONTIENE	21
2.2.1 Añadir valores a tiempos	22
3. Creación de nuestras consultas	22
4. Creación del Modelo ER	25

1.1 CREACION DE TABLAS

create table chefs

(id_chef number(2) constraint chefs_id_chefs_pk primary key,
nombre varchar2(25) constraint chefs_nombre_nn not null,
estrellas number(1) constraint chefs_estrellas_nn not null);

create table recetas

(id_recetas number(2) constraint recetas_id_recetas_pk primary key,
nombre varchar2(25) constraint recetas_nombre_nn not null,
calorias number(5) constraint recetas_calorias_nn not null,
comensales number(2) constraint recetas_comensales_nn not null,
categoria varchar2(25) constraint recetas_categoria_nn not null);

create table ingredientes

(id_ing number(2) constraint ingredientes_id_ing_pk primary key,
nombre varchar2(25) constraint ingredientes_nombre_nn not null,
tipo varchar2(25) constraint ingredientes_tipo_nn not null);

create table usuarios

(user_id number(2) constraint usuarios_user_id_nn not null,
nombre varchar2(25) constraint usuarios_nombre_nn not null,
constraint user_pk primary key (user_id));

create table evaluacion(

estrellas number(1) constraint evaluacion_estrellas_ck check (estrellas >= 1 and estrellas <= 5),
user_id number(2) constraint evaluacion_user_id_fk references usuarios(user_id),
id_receta number(2) constraint evaluacion_id_receta_fk references recetas(id_recetas),
constraint evaluacion_pk primary key (user_id,id_receta));

create table autoria(

id_chef number(2) constraint autoria_id_chef_fk references chefs(id_chef),
id_receta number(2) constraint autoria_id_receta_fk references recetas(id_recetas),
constraint autoria_pk primary key(id_chef,id_receta));

create table contiene

(id_receta number(2) constraint contiene_id_receta_fk references recetas(id_recetas),
id_ingrediente number(2) constraint contiene_id_ing_fk references ingredientes(id_ing),
cantidad number(3) constraint contiene_cantidad_nn not null,
unidad varchar2(25) constraint contiene_unidad_nn not null,
constraint contiene_pk primary key(id_receta, id_ingrediente));

1.2 SECUENCIAS

```
create sequence chef_id_seq  
increment by 1  
start with 1;
```

```
create sequence id_receta_seq  
increment by 1  
start with 1;
```

```
create sequence id_ing_seq  
increment by 1  
start with 1;
```

```
create sequence user_id_seq  
increment by 1  
start with 1;
```

1.3 INSTERTS

CHEFS

```
insert into chefs(id_chef, nombre, estrellas)  
values (chef_id_seq.NEXTVAL,'Antone',3);
```

```
insert into chefs(id_chef, nombre, estrellas)  
values (chef_id_seq.NEXTVAL,'Maria',4);
```

```
insert into chefs(id_chef, nombre, estrellas)  
values (chef_id_seq.NEXTVAL,'Juan',5);
```

```
insert into chefs(id_chef, nombre, estrellas)  
values (chef_id_seq.NEXTVAL,'Pedro',2);
```

RECETAS

```
insert into recetas(id_recetas, nombre, calorías, comensales, categoría)
values(id_receta_seq.NEXTVAL,'Omelet', 100, 4, 'Huevos');
```

```
insert into recetas(id_recetas, nombre, calorías, comensales, categoría)
values(id_receta_seq.NEXTVAL,'Guisaldo', 400, 2, 'Carne');
```

```
insert into recetas(id_recetas, nombre, calorías, comensales, categoría)
values(id_receta_seq.NEXTVAL,'Paella',500, 4, 'Arroces');
```

```
insert into recetas(id_recetas, nombre, calorías, comensales, categoría)
values(id_receta_seq.NEXTVAL,'Carne Tierna', 400, 3, 'Carnes');
```

```
insert into recetas(id_recetas, nombre, calorías, comensales, categoría)
values(id_receta_seq.NEXTVAL,'Sopa Tomate', 150, 4, 'Sopas');
```

```
insert into recetas(id_recetas, nombre, calorías, comensales, categoría)
values(id_receta_seq.NEXTVAL,'Crepes', 270, 4, 'Postres');
```

```
insert into recetas(id_recetas, nombre, calorías, comensales, categoría)
values(id_receta_seq.NEXTVAL,'Carne Porcina Especial', 300, 4, 'Carnes');
```

```
insert into recetas(id_recetas, nombre, calorías, comensales, categoría)
values(id_receta_seq.NEXTVAL,'Asado', 400, 3, 'Carnes');
```

```
insert into recetas(id_recetas, nombre, calorías, comensales, categoría)
values(id_receta_seq.NEXTVAL,'Helado', 330, 4, 'Postres');
```

INGREDIENTES

```
insert into ingredientes(id_ing, nombre, tipo)
values (id_ing_seq.NEXTVAL, 'tomate', 'vegetal');
```

```
insert into ingredientes(id_ing, nombre, tipo)
values (id_ing_seq.NEXTVAL, 'huevo', 'huevo');
```

```
insert into ingredientes(id_ing, nombre, tipo)
values (id_ing_seq.NEXTVAL, 'leche', 'lacteo');
```

```
insert into ingredientes(id_ing, nombre, tipo)
values (id_ing_seq.NEXTVAL, 'carne vacuna', 'vacuno');
```

```
insert into ingredientes(id_ing, nombre, tipo)
values (id_ing_seq.NEXTVAL, 'queso', 'lacteo');
```

```
insert into ingredientes(id_ing, nombre, tipo)
values (id_ing_seq.NEXTVAL, 'pollo', 'ave');
```

```
insert into ingredientes(id_ing, nombre, tipo)
values (id_ing_seq.NEXTVAL, 'carne porcina', 'porcino');
```

```
insert into ingredientes(id_ing, nombre, tipo)
values (id_ing_seq.NEXTVAL, 'arroz', 'arroz');
```

```
insert into ingredientes(id_ing, nombre, tipo)
values (id_ing_seq.NEXTVAL, 'fettuccini', 'pasta');
```

```
insert into ingredientes(id_ing, nombre, tipo)
values (id_ing_seq.NEXTVAL, 'nata liquida', 'lacteo');
```

```
insert into ingredientes(id_ing, nombre, tipo)
values (id_ing_seq.NEXTVAL, 'nata p/montar', 'lacteo');
```

```
insert into ingredientes(id_ing, nombre, tipo)
values (id_ing_seq.NEXTVAL, 'azucar blanco', 'azucar');
```

```
insert into ingredientes(id_ing, nombre, tipo)
values (id_ing_seq.NEXTVAL, 'azucar moreno', 'azucar');
```

```
insert into ingredientes(id_ing, nombre, tipo)
values (id_ing_seq.NEXTVAL, 'vainilla', 'vegetal');
```

USUARIOS

```
insert into usuarios(user_id, nombre)
values (user_id_seq.NEXTVAL, 'Pere');
```

```
insert into usuarios(user_id, nombre)
values (user_id_seq.NEXTVAL, 'Marcela');
```

```
insert into usuarios(user_id, nombre)
values (user_id_seq.NEXTVAL, 'Ali');
```

```
insert into usuarios(user_id, nombre)
values (user_id_seq.NEXTVAL, 'Franca');
```

```
insert into usuarios(user_id, nombre)
values (user_id_seq.NEXTVAL, 'John');
```

EVALUACION

```
insert into evaluacion(user_id, id_receta, estrellas)
values (1,1,4);
```

```
insert into evaluacion(user_id, id_receta, estrellas)
values (1,2,2);
```

```
insert into evaluacion(user_id, id_receta, estrellas)
values (2,1,4);
```

```
insert into evaluacion(user_id, id_receta, estrellas)
values (2,3,5);
```

```
insert into evaluacion(user_id, id_receta, estrellas)
values (4,4,5);
```

```
insert into evaluacion(user_id, id_receta, estrellas)
values (5,4,5);
```

```
insert into evaluacion(user_id, id_receta, estrellas)
values (3,8,3);
```

AUTORIA

```
insert into autoria(id_chef,id_receta)
values (1,1);
```

```
insert into autoria(id_chef,id_receta)
values (1,2);
```

```
insert into autoria(id_chef,id_receta)
values (2,3);
```

```
insert into autoria(id_chef,id_receta)
values (3,4);
```

```
insert into autoria(id_chef,id_receta)
values (4,5);
```

```
insert into autoria(id_chef,id_receta)
values (4,6);
```

```
insert into autoria(id_chef,id_receta)
values (1,7);
```

```
insert into autoria(id_chef,id_receta)
values (1,8);
```

```
insert into autoria(id_chef,id_receta)
values (2,9);
```


CONTIENE

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(1,2,200,'gr');
```

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(1,3,100,'ml');
```

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(2,4,50,'gr');
```

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(2,1,70,'gr');
```

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(3,6,100,'ml');
```

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(3,1,150,'ml');
```

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(5,1,200,'gr');
```

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(5,3,100,'gr');
```

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(6,3,50,'ml');
```

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(6,2,100,'gr');
```

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(4,4,10,'gr');
```

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(4,3,100,'gr');
```

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(7,7,200,'gr');
```

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(8,4,400,'gr');
```

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(8,1,100,'gr');
```

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(7,1,50,'gr');
```

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(9,14,1,'unidad');
```

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(9,11,500,'ml');
```

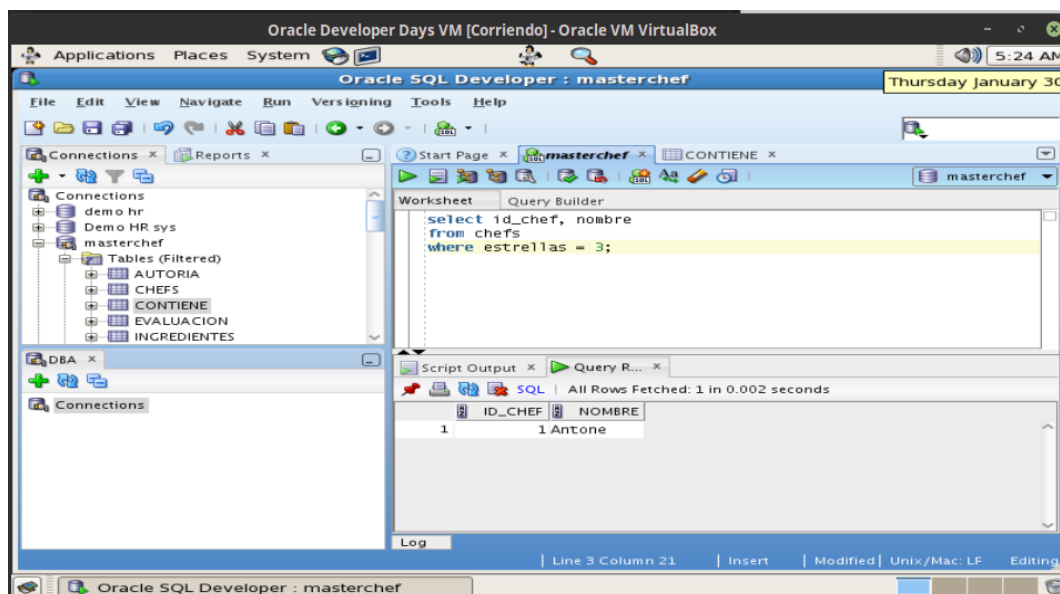
```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(9,12,200,'gr');
```

```
insert into contiene(id_receta, id_ingredient, cantidad, unidad)
values(3,8,500,'gr');
```

1.4 CONSULTAS

1. Listar todos los chefs que tienen 3 estrellas.

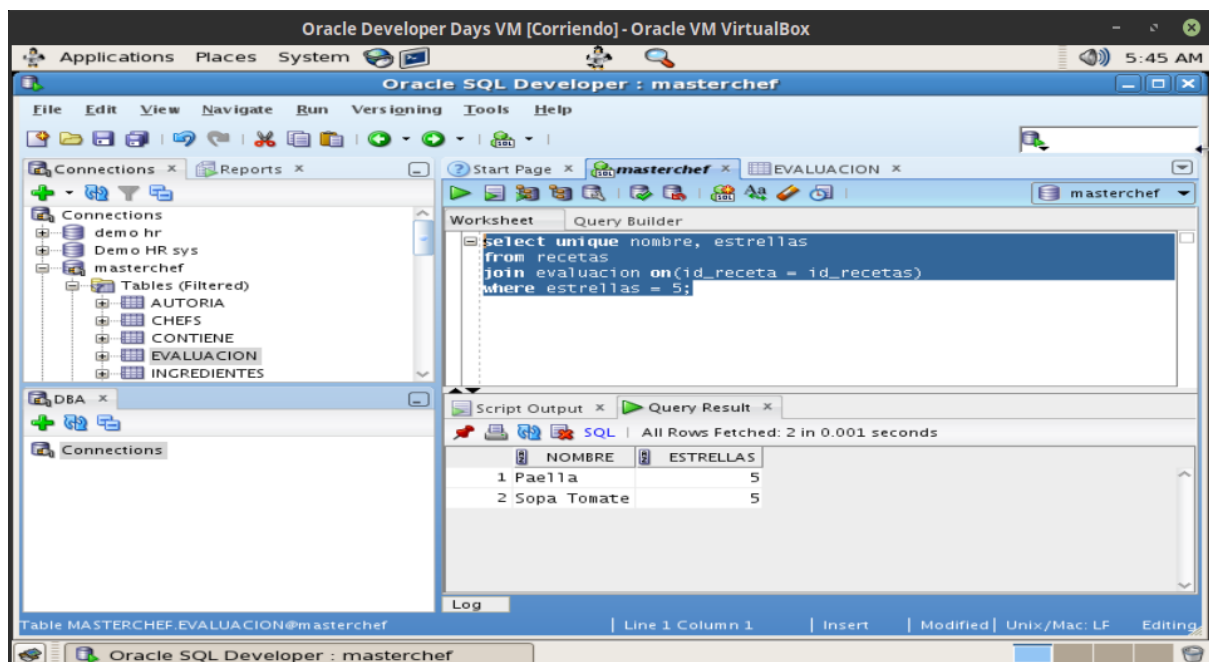
```
select id_chef, nombre
from chefs
where estrellas = 3;
```



2. Listar los nombres de las recetas que tienen al menos una evaluación 5 estrellas. Sin repetir los nombres de las recetas.

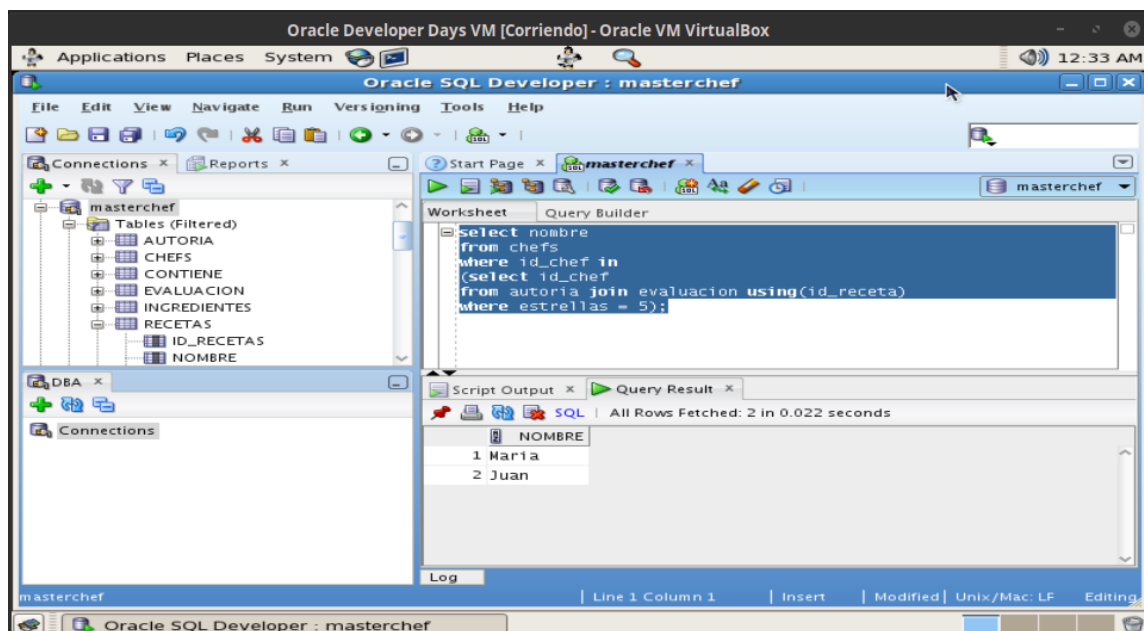
```
select unique nombre, estrellas
from recetas
join evaluacion on(id_receta = id_recetas)
where estrellas = 5;
```

en la tabla recetas hay una columna que se llama id_recetas y en el resto la hemos llamado id_receta por esto hemos tenido algunos problemas y por eso el join es a si



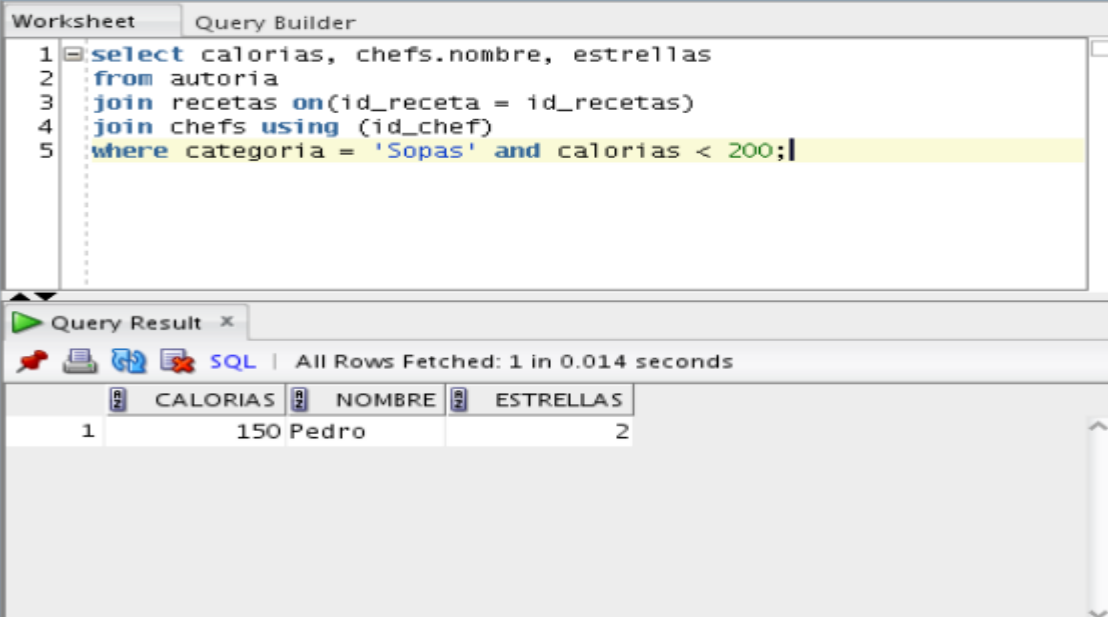
3. Listar todos los chefs que tienen al menos un comentario de una receta con 5 estrellas.

```
select nombre  
from chefs  
where id_chef in  
(select id_chef  
from autoria join evaluacion using(id_receta)  
where estrellas = 5);
```



4. Listar las sopas con menos de 200 calorías por persona, para cada sopa listada se desea saber el nombre del chef que creó la sopa y cuantas estrellas tiene ese chef.

```
select unique calorías, chefs.nombre, estrellas
from autoria
join recetas on(id_receta = id_recetas)
join chefs using (id_chef)
where categoria = 'Sopas' and calorías < 200;
```

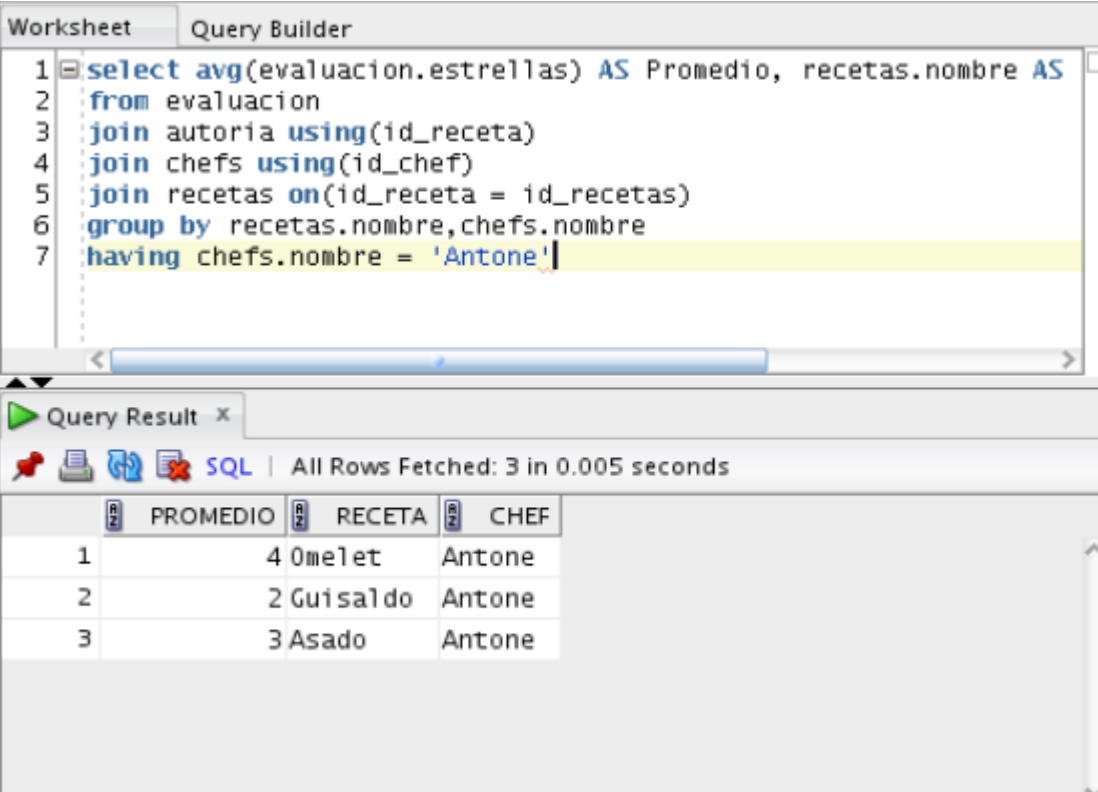


The screenshot shows a database query builder interface. The top section, titled 'Query Builder', contains a SQL query. The bottom section, titled 'Query Result', shows the results of the query. The query is: `select unique calorías, chefs.nombre, estrellas from autoria join recetas on(id_receta = id_recetas) join chefs using (id_chef) where categoria = 'Sopas' and calorías < 200;`. The results table has three columns: CALORIAS, NOMBRE, and ESTRELLAS. The first row shows 150 for CALORIAS, Pedro for NOMBRE, and 2 for ESTRELLAS.

	CALORIAS	NOMBRE	ESTRELLAS
1	150	Pedro	2

5. Cual es el promedio de estrellas de cada receta que creó el chef "Antone"

```
select avg(evaluacion.estrellas) AS Promedio, recetas.nombre AS Receta, chefs.nombre AS Chef
from evaluacion
join autoria using(id_receta)
join chefs using(id_chef)
join recetas on(id_receta = id_recetas)
group by recetas.nombre,chefs.nombre
having chefs.nombre = 'Antone';
```



The screenshot displays a database query builder interface. The top section, labeled 'Query Builder', contains the following SQL query:

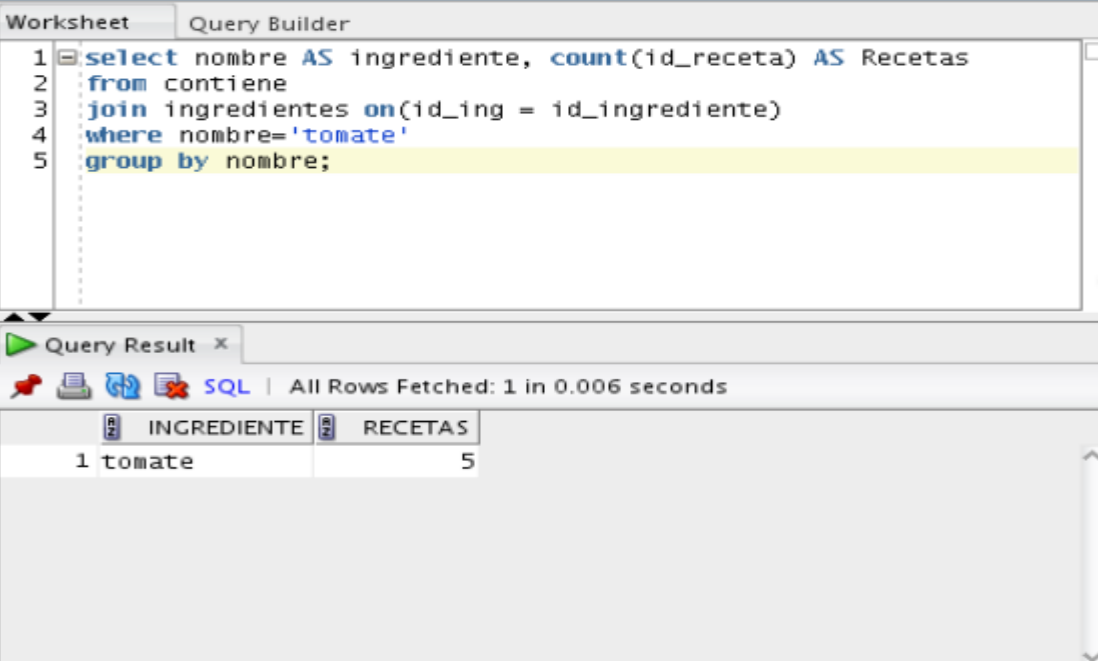
```
1 select avg(evaluacion.estrellas) AS Promedio, recetas.nombre AS
2 from evaluacion
3 join autoria using(id_receta)
4 join chefs using(id_chef)
5 join recetas on(id_receta = id_recetas)
6 group by recetas.nombre,chefs.nombre
7 having chefs.nombre = 'Antone';
```

Below the query builder is the 'Query Result' section, which shows the results of the query. It indicates that all rows were fetched in 0.005 seconds. The results are displayed in a table with three columns: PROMEDIO, RECETA, and CHEF.

	PROMEDIO	RECETA	CHEF
1	4	Omelet	Antone
2	2	Guisado	Antone
3	3	Asado	Antone

6. Cuantas recetas usan tomate?

```
select nombre AS ingrediente, count(id_receta) AS Recetas
from contiene
join ingredientes on(id_ing = id_ingrediente)
where nombre='tomate'
group by nombre;
```



The screenshot shows a database query builder interface. The top pane, titled 'Query Builder', contains the following SQL query:

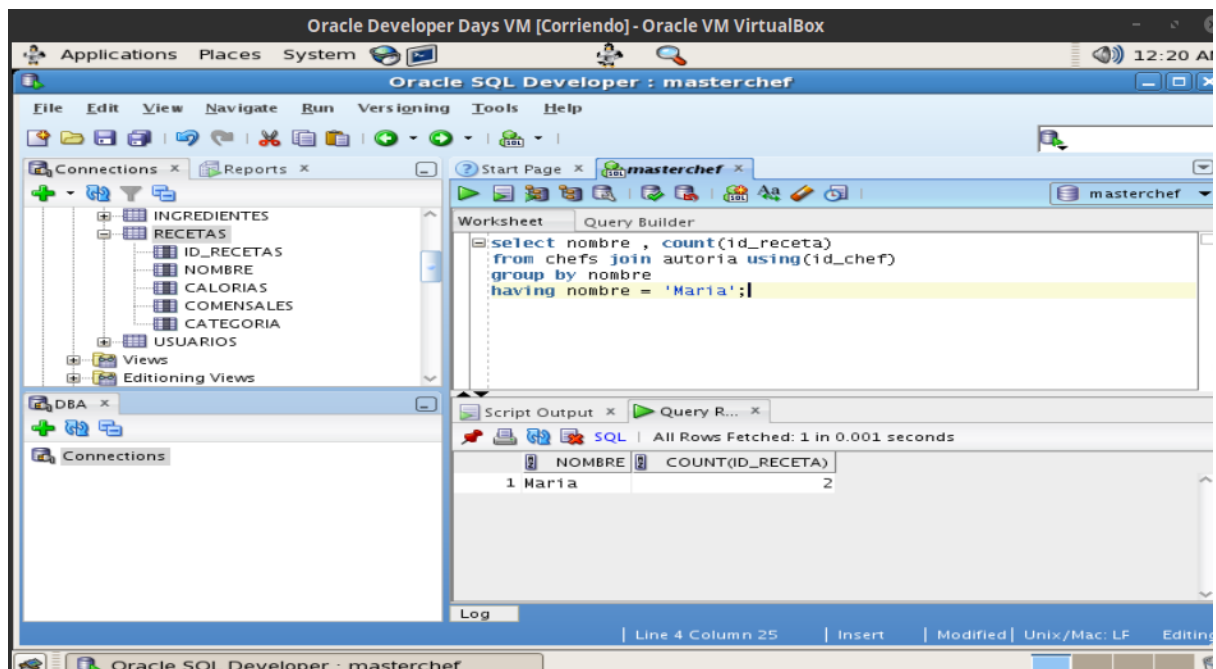
```
1 select nombre AS ingrediente, count(id_receta) AS Recetas
2 from contiene
3 join ingredientes on(id_ing = id_ingrediente)
4 where nombre='tomate'
5 group by nombre;
```

The bottom pane, titled 'Query Result', shows the results of the query. It includes a toolbar with icons for saving, refreshing, and other actions, along with the text 'SQL | All Rows Fetched: 1 in 0.006 seconds'. The results are displayed in a table with two columns: 'INGREDIENTE' and 'RECETAS'.

INGREDIENTE	RECETAS
1 tomate	5

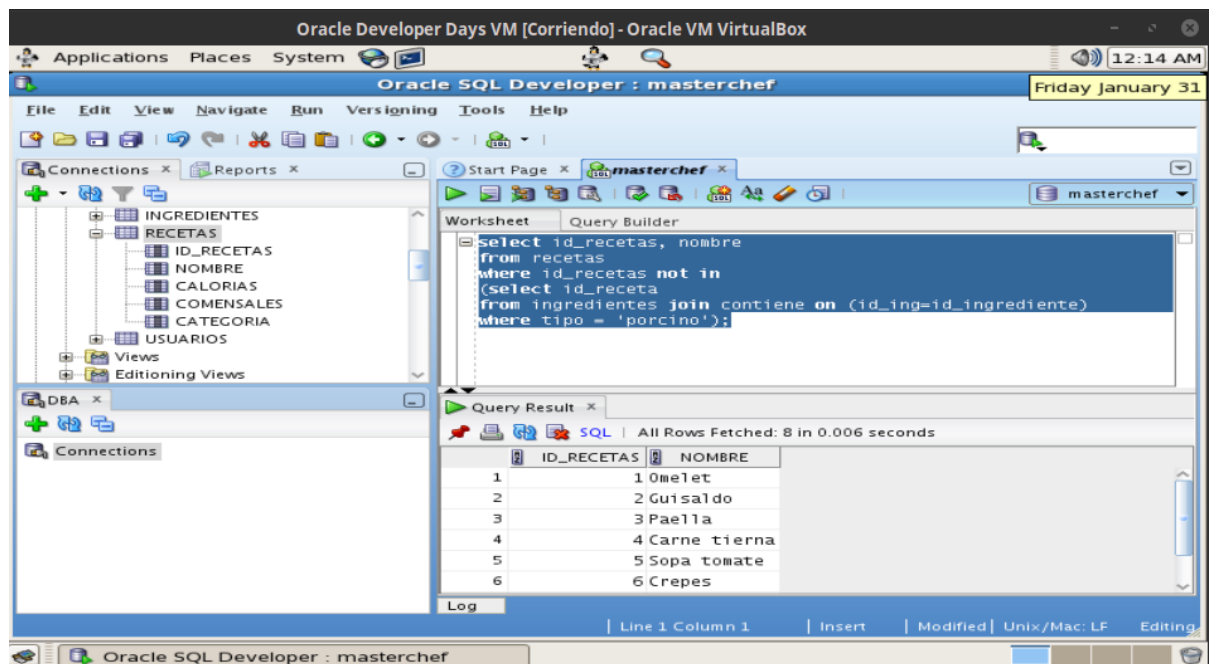
7. Cuantas recetas creo "Maria"?

```
select nombre , count(id_receta)
from chefs join autoria using(id_chef)
group by nombre
having nombre = 'Maria';
```



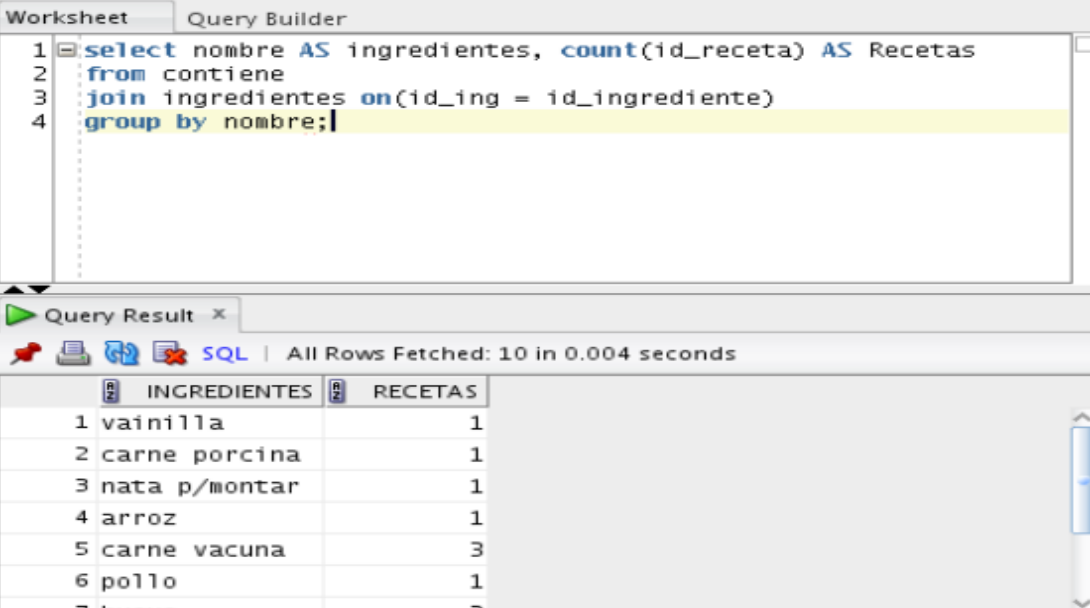
8. Listar los nombres de las recetas que no llevan carne porcina (carne porcina o porcino es un tipo de ingrediente)

```
select id_recetas, nombre
from recetas
where id_recetas not in
(select id_receta
from ingredientes join contiene on (id_ing=id_ingrediente)
where tipo = 'porcino');
```



9. Para cada ingrediente se desea saber cuántas recetas lo usan.

```
select nombre AS ingredientes, count(id_receta) AS Recetas
from contiene
join ingredientes on(id_ing = id_ingrediente)
group by nombre;
```



Worksheet Query Builder

```
1 select nombre AS ingredientes, count(id_receta) AS Recetas
2 from contiene
3 join ingredientes on(id_ing = id_ingrediente)
4 group by nombre;
```

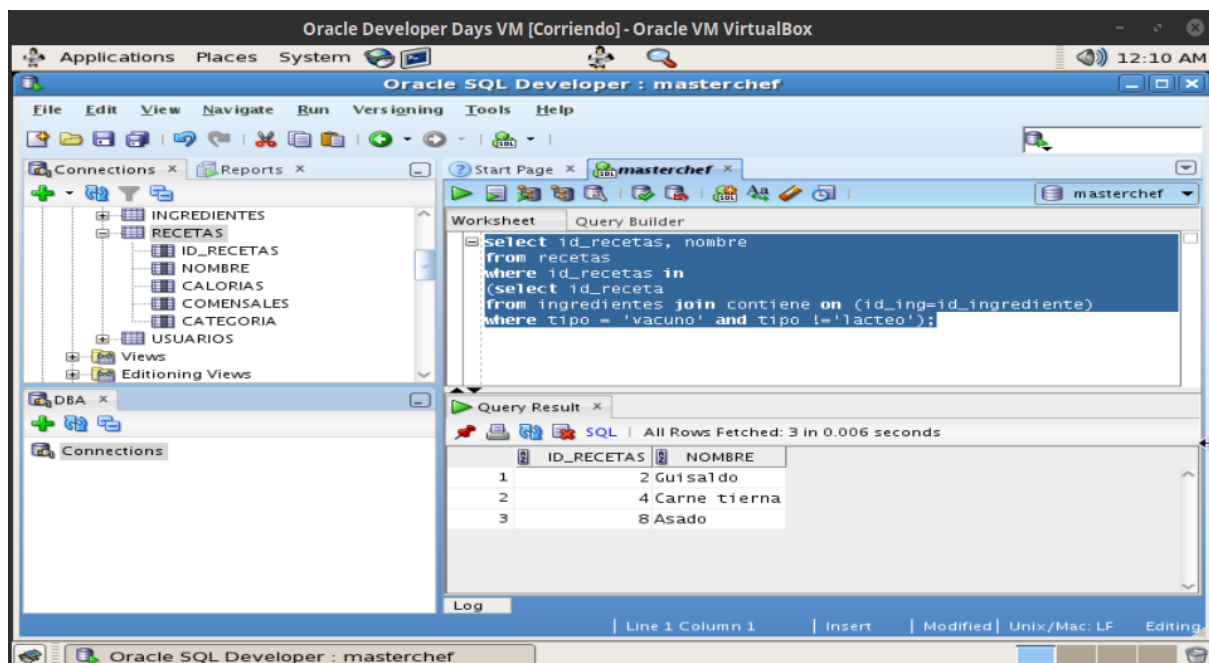
Query Result x

SQL | All Rows Fetched: 10 in 0.004 seconds

	INGREDIENTES	RECETAS
1	vainilla	1
2	carne porcina	1
3	nata p/montar	1
4	arroz	1
5	carne vacuna	3
6	pollo	1
7	huevo	2

10. Tengo una amiga a cenar a quien le gusta la carne vacuna pero es alérgica a los lácteos escribir un comando SQL para recomendar recetas tales que contengan carne vacuna pero no contenga ingredientes que le dan alergia (como categoría de ingredientes).

```
select id_recetas, nombre
from recetas
where id_recetas in
(select id_receta
from ingredientes join contiene on (id_ing=id_ingrediente)
where tipo = 'vacuno' and tipo !='lacteo');
```



2. APARTADOS EXTRAS

2.1 Creación de nueva tabla

```
create table tiempos
(id_receta number(2) constraint tiempos_id_receta_fk references
recetas(id_recetas),
tiempo number(4) constraint tiempos_tiempo_nn not null,
unidad varchar2(25) constraint tiempos_unidad_nn not null,
constraint tiempos_pk primary key(id_receta));
```

2.2 Datos adicionales

CHEFS

```
insert into chefs(id_chef, nombre, estrellas)
values (5,'Dario',1);
```

RECETAS

```
insert into recetas(id_recetas, nombre, calorías, comensales, categoría)
values(10,'Sopa de pollo', 250, 4, 'Sopas');
```

INGREDIENTES

```
insert into ingredientes(id_ing, nombre, tipo)
values (15, 'Zanahoria', 'vegetal');
```

USUARIOS

```
insert into usuarios(user_id, nombre)
values (6, 'Marco');
```

EVALUACIÓN

```
insert into evaluacion(user_id, id_receta, estrellas)
values (6,10,5);
```

```
insert into evaluacion(user_id, id_receta, estrellas)
values (6,6,5);
```

AUTORÍA

```
insert into autoria(id_chef,id_receta)
values (5,10);
```

CONTIENE

```
insert into contiene(id_receta, id_ingrediente, cantidad, unidad)
values(10,15,80,'gr');
```

```
insert into contiene(id_receta, id_ingrediente, cantidad, unidad)
values(10,6,200,'gr');
```

2.2.1 Añadir valores a tiempos

```
insert into tiempos(id_receta, tiempo, unidad)
values(1,15,'minutos');
```

```
insert into tiempos(id_receta, tiempo, unidad)
values(2,45,'minutos');
```

```
insert into tiempos(id_receta, tiempo, unidad)
values(3,1,'hora');
```

```
insert into tiempos(id_receta, tiempo, unidad)
values(4,40,'minutos');
```

```
insert into tiempos(id_receta, tiempo, unidad)
values(5,25,'minutos');
```

```
insert into tiempos(id_receta, tiempo, unidad)
values(6,130,'segundos');
```

```
insert into tiempos(id_receta, tiempo, unidad)
values(7,35,'minutos');
```

```
insert into tiempos(id_receta, tiempo, unidad)
values(8,1,'hora');
```

```
insert into tiempos(id_receta, tiempo, unidad)
values(9,10,'minutos');
```

```
insert into tiempos(id_receta, tiempo, unidad)
values(10,25,'minutos');
```

3. Creación de nuestras consultas

1. Cuanto tarda el chef “maria” en preparar sus platos?

```
SELECT chefs.nombre AS Chef, recetas.nombre AS Receta, tiempo, unidad from tiempos
join recetas on (id_receta = id_recetas)
join autoria using (id_receta)
join chefs using (id_chef)
where chefs.nombre = 'Maria';
```

The screenshot shows a database query builder interface. The top section, labeled 'Query Builder', contains the following SQL query:

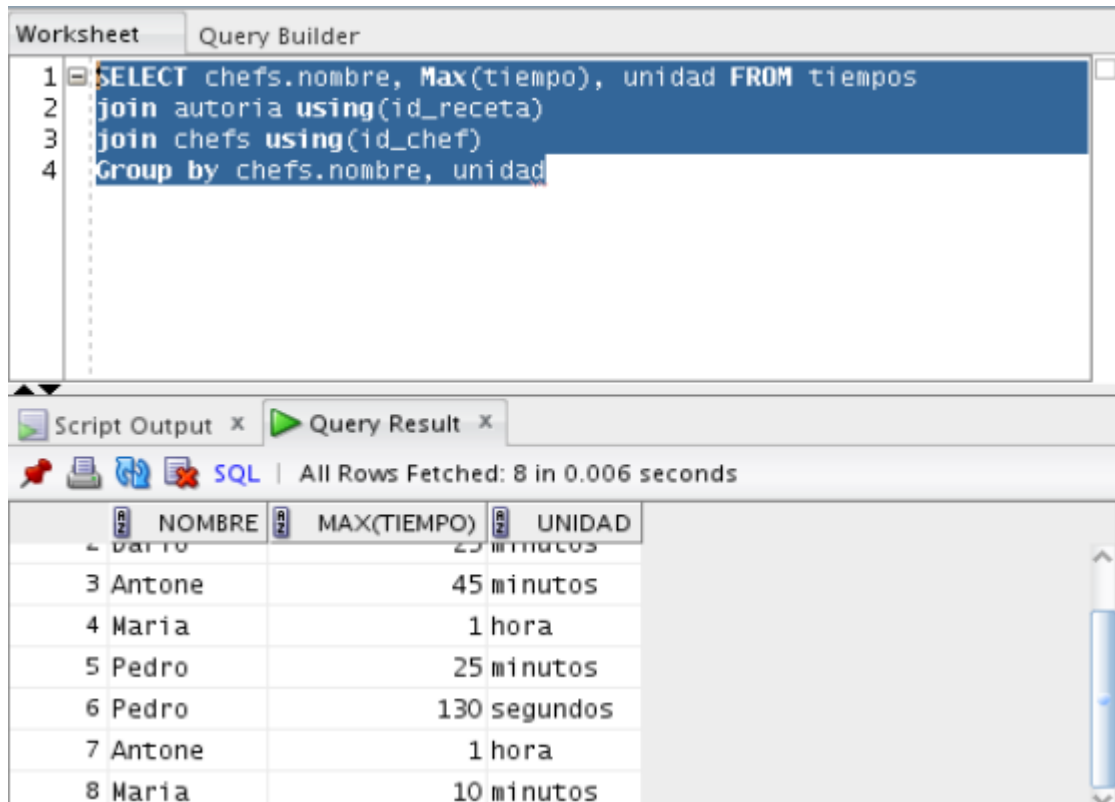
```
1 SELECT chefs.nombre AS Chef, recetas.nombre AS Receta, tiempo, u
2 join recetas on (id_receta = id_recetas)
3 join autoria using (id_receta)
4 join chefs using (id_chef)
5 where chefs.nombre = 'Maria';
```

Below the query editor, the 'Query Result' tab is active, displaying the results of the query. The status bar indicates 'All Rows Fetched: 2 in 0.003 seconds'. The results are shown in a table with the following columns: CHEF, RECETA, TIEMPO, and UNIDAD.

	CHEF	RECETA	TIEMPO	UNIDAD
1	Maria	Paella	1 hora	
2	Maria	Helado	10 minutos	

2. Muestra el tiempo Máximo de receta y los chefs en todos los tiempos (hora,minutos,segundos)

```
SELECT chefs.nombre, Max(tiempo), unidad FROM tiempos  
join autoria using(id_receta)  
join chefs using(id_chef)  
Group by chefs.nombre, unidad;
```



The screenshot shows a database query builder interface. The top section, labeled 'Query Builder', contains the following SQL query:

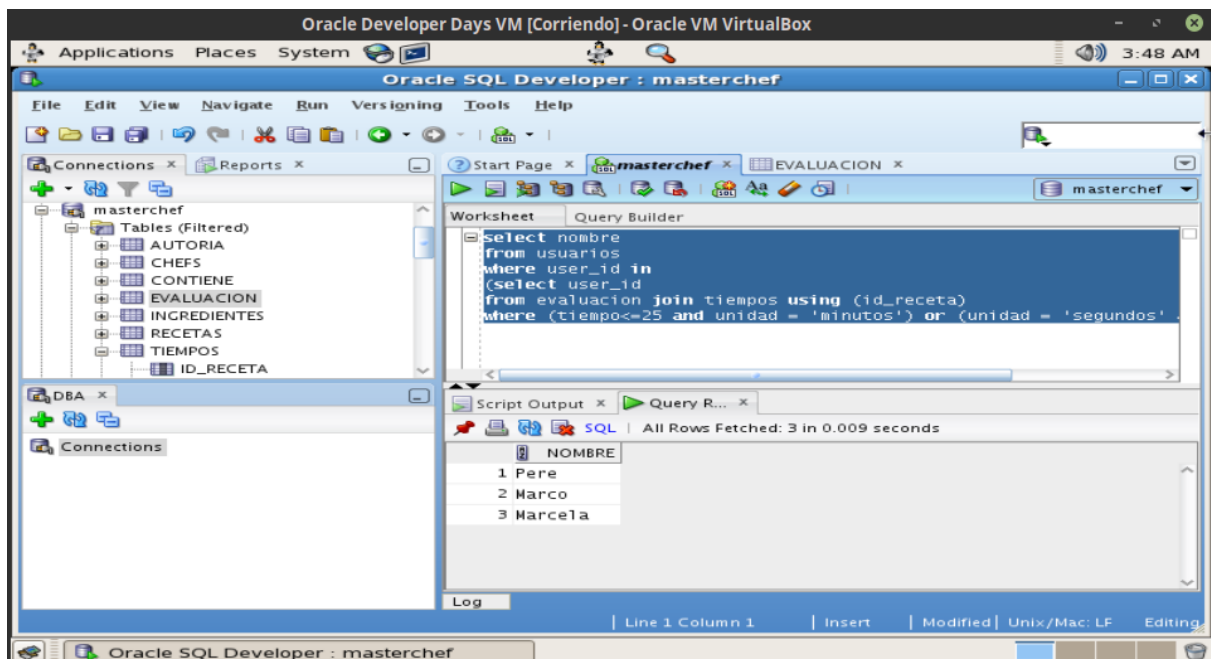
```
1 SELECT chefs.nombre, Max(tiempo), unidad FROM tiempos  
2 join autoria using(id_receta)  
3 join chefs using(id_chef)  
4 Group by chefs.nombre, unidad;
```

Below the query builder, there is a 'Query Result' tab. It shows the results of the query, indicating that all 8 rows were fetched in 0.006 seconds. The results are displayed in a table with three columns: NOMBRE, MAX(TIEMPO), and UNIDAD.

NOMBRE	MAX(TIEMPO)	UNIDAD
Antone	45 minutos	
Maria	1 hora	
Pedro	25 minutos	
Pedro	130 segundos	
Antone	1 hora	
Maria	10 minutos	

3. Muestra los usuarios que hayan evaluado las recetas que tarden 25 minutos o menos

```
select nombre
from usuarios
where user_id in
(select user_id
from evaluacion join tiempos using (id_receta)
where (tiempo<=25 and unidad = 'minutos') or (unidad = 'segundos' and tiempo<=1500));
```



4. Creación del Modelo ER

