

Tutorial 3 – SQL Part II

Inner Joins and Simple Subqueries

WITH SUGGESTED SOLUTIONS

Solutions to Review Questions

1. What is the difference between a natural join and an inner join?

Ans: If more than one table is listed in the FROM clause of a SELECT statement, and no join condition between the tables is specified, the database engine will try to perform a natural join, i.e. it will check if there are columns in each table that have exactly the same name, and if this is true, it will perform the join between the tables with these columns, checking matching values for each row from the tables.

However, it is always best practice to have a join condition explicitly specified, either as an INNER JOIN, or simply JOIN clause in the FROM clause, or in a WHERE condition, to be sure which columns from the tables will be used in the join operation. This is because if the database engine cannot find common column names in both of the tables, it will perform what is called an 'extended cross-product' (see answer below for d.). In the small tables that we've used in this subject, you can usually see at a glance if there are common columns, but in most business organisations, there are tables with dozens of columns, and it may be very difficult to ascertain if there are common columns. You don't really want to discover that there are no common columns between tables by crashing the database server as it tries to execute your query without a specific inner join condition!

The moral of the story is: do not ever have a SELECT statement that queries more than one table without explicitly coding the join condition, relying on natural joins is a big No-No!

2. What is the difference between a (natural or inner) join operation and a simple subquery operation?

Ans: Join – relational operation that causes two tables with a common domain to be combined into a single table. Every row in each table is compared with every row in the second table to test whether or not to combine these rows; only rows with matching values are included in the result.

Subquery – involves placing a SELECT query within a WHERE or HAVING clause of another, outer query. The inner or nested query provides a set of one or more values to be used in the search condition of the outer query. The nested query is only executed once, while the outer query processes every row in the table(s) in its FROM clause and uses the results of the nested query to decide which values to display. Only the results from the outer query table(s) are actually displayed.

Solutions to Problems and Exercises

The following SQL exercises are based on the Pizza database that consists of 3 tables:

MENU(pizza, price, country, base)

ITEMS(ingredient, type)

RECIPE(pizza, ingredient, amount)

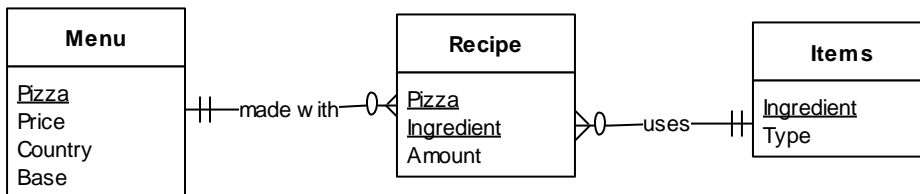
Each row in the MENU table is uniquely identified by the pizza name (PIZZA), and contains the price (PRICE), the country of origin (COUNTRY), and the base used in making the pizza (BASE), i.e. whole meal (wm) or white flour (wf). The ITEMS table records the ingredients used for making the pizzas. Each unique ingredient (INGREDIENT) has a type (TYPE) to indicate if the ingredient is dairy, meat, fish, etc. Finally, RECIPE table records the combinations of pizzas and the ingredients used in making each pizza, also specifying the amount in grams (AMOUNT) for each main ingredient.

```
create table menu
(
    pizza      varchar(20),
    price      real,
    country    varchar(20),
    base       varchar(20),
    PRIMARY KEY (pizza)
);

create table items
(
    ingredient  varchar(12),
    type       varchar(8),
    PRIMARY KEY (ingredient)
);

create table recipe
(
    pizza      varchar(20),
    ingredient  varchar(12),
    amount     integer,
    PRIMARY KEY (pizza,ingredient),
    FOREIGN KEY (pizza) REFERENCES menu,
    FOREIGN KEY (ingredient) REFERENCES items);
```

Entity-Relationship Diagram



SECTION ONE

For each question below, do the following:

- Construct a SELECT statement that contains an **Inner Join**.
- Manually trace through the data using your SELECT statement to verify that it is correct, and that you understand exactly how the statement is executed.
- Construct a SELECT statement that contains a Simple Subquery.
- Manually trace the execution of the Simple Subquery statement
- Describe the differences in execution steps and the results between using an Inner Join and a Simple Subquery to provide the information requested in the question.

1. List all ingredients and their type used in the Margarita pizza.

a)
 select items.ingredient, type
 from recipe, items
 where pizza = 'margarita'
 and items.ingredient = recipe.ingredient;

b)
 select items.ingredient, type
 from recipe natural join items
 where pizza = 'margarita';

c) simple subquery:

```
select ingredient, type
From items
Where ingredient in (select ingredient
                    From recipe
                    Where pizza = 'margarita');
```

Result

```
ingredient | type
-----+-----
cheese     | dairy
spice      | spice
(2 rows)
```

2. List fish ingredients used in pizzas.

a)
 select pizza, items.ingredient
 from items natural join recipe
 where type = 'fish'
 order by pizza;

b)
 select pizza, items.ingredient
 from items join recipe
 on items.ingredient = recipe.ingredient
 where type = 'fish'
 order by pizza;

```
pizza      | ingredient
-----+-----
napolitana | anchovies
seafood    | seafood
siciliano  | anchovies
special    | anchovies
special    | seafood
stagiony   | anchovies
(6 rows)
```

c) simple subquery:

```
select pizza,ingredient
From recipe
      Where ingredient in (select ingredient
                          From items
                          Where type = 'fish');
```

Result

pizza	ingredient
napolitana	anchovies
siciliano	anchovies
seafood	seafood
special	anchovies
special	seafood
stagiony	anchovies

(6 rows)

3. Give all pizzas that originate from the same country as the Siciliano pizza.

```
select m1.pizza
from menu m1, menu m2
where m1.country = m2.country
and m2.pizza = 'siciliano'
and m1.pizza <> 'siciliano';
```

```
select pizza
from menu
where pizza <> 'siciliano' and
      country = (select country from menu where pizza='siciliano');
```

Result

pizza
margarita
napolitana
stagiony
cabanossi

(4 rows)

4. List all pizzas that cost more than the Stagiony pizza; list pizza name and price.

```
select m1.pizza, m1.price
from menu m1, menu m2
where m1.price > m2.price
and m2.pizza = 'stagiony';
```

```
select pizza, price
From menu
```

Where price > (select price
 From menu
 Where pizza = 'stagiony');

Result

```

pizza | price
-----+-----
special | 9.9
seafood | 9.2
(2 rows)

```

5. For the Americano pizza give the base, and list all ingredients and their type.

a) select m.pizza, m.base, i.ingredient, i.type
 from menu m, recipe r, items i
 where m.pizza = r.pizza
 and r.ingredient = i.ingredient
 and m.pizza = 'americano';

b) select pizza, base, ingredient, type from menu, items
 where
 items.ingredient in (select ingredient from recipe where pizza = 'americano')
 and
 pizza = 'americano';

Result

```

pizza | base | ingredient | type
-----+-----+-----+-----
americano | Wm | salami | meat
americano | Wm | pepperoni | meat
americano | Wm | spice | spice
(3 rows)

```

6. Give pizzas with price that falls between the price of Garlic and Ham pizzas.

a) select m1.pizza, m1.price
 from menu m1, menu m2, menu m3
 where m1.price between m2.price and m3.price
 and m2.pizza = 'garlic'
 and m3.pizza = 'ham'
 order by 2;

Note : Using BETWEEN will cover the price of garlic and ham (equivalent to <= and >=)

Tut 4 →

Q7) SELECT m1.pizza, m1.price
 FROM menu m1, menu m2, menu m3
 WHERE m1.price between m2.price and m3.price
 and m2.pizza = 'garlic'
 and m3.pizza = 'ham'

SELECT pizza, price
 FROM menu
 WHERE price between (
 SELECT price from menu where pizza = 'garlic')
 and (
 SELECT price from menu where pizza = 'ham');

① MENU m1

pizza	price
garlic	3.5
ham	7.3
me	
margherita	6.2
garlic	3.5
...	...

② MENU m2

pizza	price
margherita	6.2
<u>garlic</u>	<u>3.5</u>
ham	7.3

menu m3

pizza	price
margherita	6.2
garlic	3.5
<u>ham</u>	<u>7.3</u>

pizza	price
garlic	3.5
margherita	6.2
ham	7.3

①

Menu	price
	<u>3.5</u>

price
<u>7.3</u>

pizza	price
garlic	3.5
margherita	6.2
garlic	7.3

```
b) select m1.pizza, m1.price
   from menu m1
   where m1.price > (select price from menu where pizza='garlic')
   and m1.price < (select price from menu where pizza='ham');
```

Note: Using < and > will not cover the price of garlic or the price of ham

Result

```
pizza | price
-----+-----
margarita | 6.2
(1 row)
```

7.

pizza	price
garlic	3.5

pizza	price
ham	7.3

pizza	price
margarita	6.2

8.

pizza	country
mexicano	Mexico

pizza	ingredient
mexicano	Onion
mexicano	capsicum
mexicano	Mushroom
mexicano	chilli
mexicano	spice

7. List all ingredients for the Mexican pizza, (COUNTRY = 'mexico').

```
a) select ingredient
from recipe
where pizza = (select pizza
               from menu
               where country = 'mexico');
```

b) alternative solution using IN predicate:

```
select ingredient
from recipe
where pizza in (select pizza
               from menu
               where country = 'mexico');
```

c) inner join:

```
select ingredient from recipe r, menu m where r.pizza=m.pizza and m.country = 'mexico';
```

Result

```
ingredient
```

```
-----
```

```
onion
```

```
capsicum
```

```
mushroom
```

```
chilli
```

```
spice
```

```
(5 rows)
```

8. List all ingredients used in Italian pizzas. Organize the list alphabetically, and avoid duplicates in the result.

```
a) select distinct ingredient
from recipe
where pizza in (select pizza
               from menu
               where country = 'italy')
order by ingredient;
```

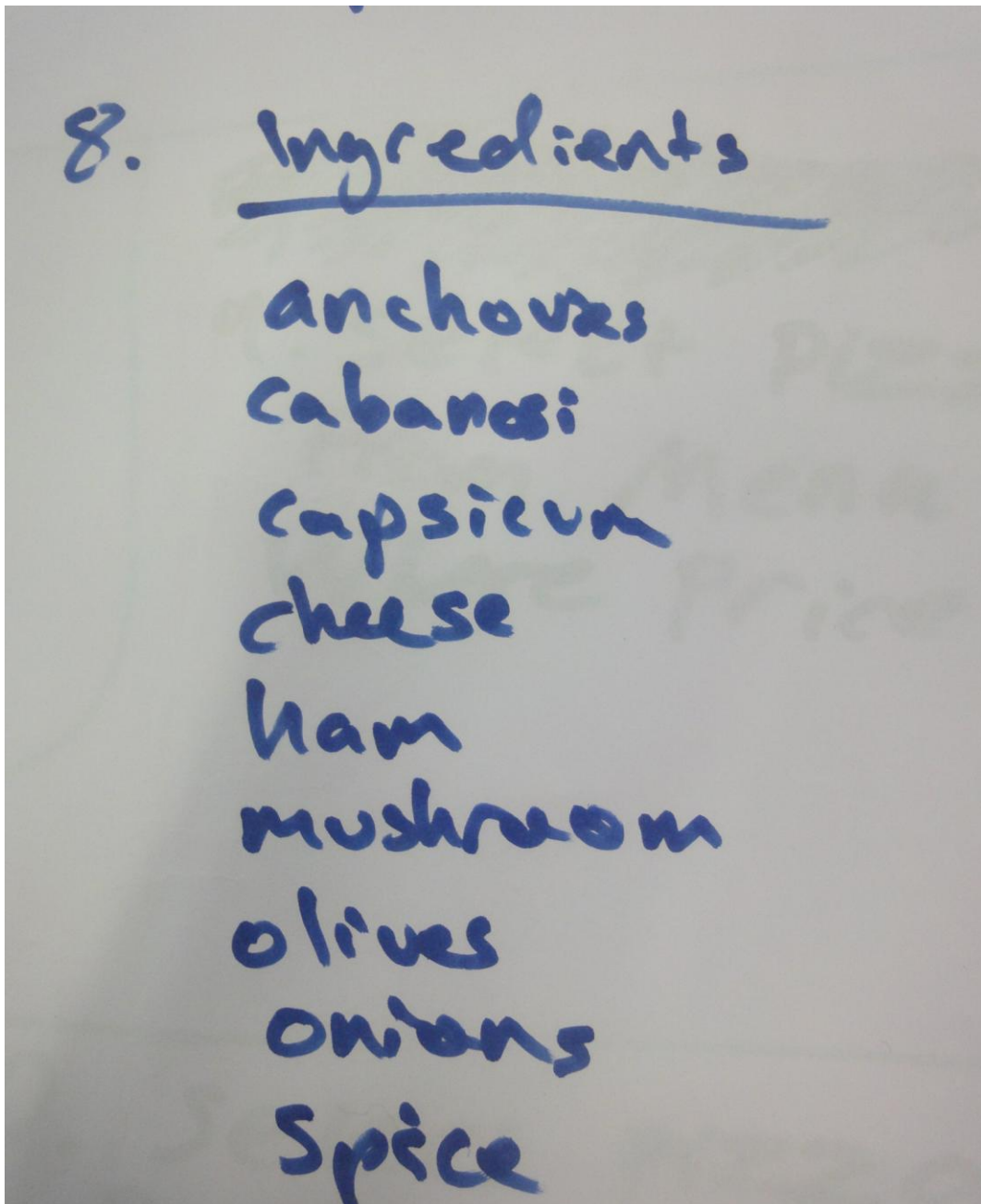
b) inner join:

```
select distinct ingredient from recipe r, menu m
where r.pizza=m.pizza
and m.country = 'italy'
order by ingredient;
```

Result

ingredient

anchovies
cabanossi
capsicum
cheese
ham
mushroom
olives
onion
spice
(9 rows)



SECTION TWO

For each question below,

- Construct a SELECT statement that contains a Simple Subquery, and
- Manually trace the execution of the Simple Subquery statement

9. Give pizzas and prices for pizzas that are more expensive than all Italian pizzas (Hint: use > all predicate).

```
select pizza, price
from menu
where price > all (select price
from menu
where country = 'italy');
```

Result

```
pizza | price
-----+-----
special | 9.9
seafood | 9.2
(2 rows)
```

10. List non-vegetarian pizzas (i.e. with at least one meat ingredient) (Hint: use =any predicate)

```
select distinct pizza
from recipe
where ingredient = any (select ingredient
from items
where type = 'meat');
```

Result

```
pizza
-----
americano
cabanossi
ham
hawaiian
special
stagiony
(6 rows)
```

11. Give the name and price of the most expensive pizza.

```
select pizza, price
from menu
where price = (select max(price) from menu);
```

Result

```

pizza | price
-----+-----
special | 9.9
(1 row)

```

12. List pizzas costing less than the average price.

```

select pizza
from menu
where price < (select avg(price) from menu);

```

Result

```

pizza
-----
margarita
ham
mushroom
garlic
(4 rows)

```

13. Give the pizza which contains the largest amount of the spice ingredient.

```

select pizza
from recipe
where amount = (select max(amount)
from recipe
where ingredient = 'spice')
and ingredient = 'spice';

```

Result

```

pizza
-----
mexicano
(1 row)

```

14. Find the pizza which uses the largest number of ingredients.

```

select pizza
from recipe
group by pizza
having count(*) >= all (select count(*)
from recipe
group by pizza);

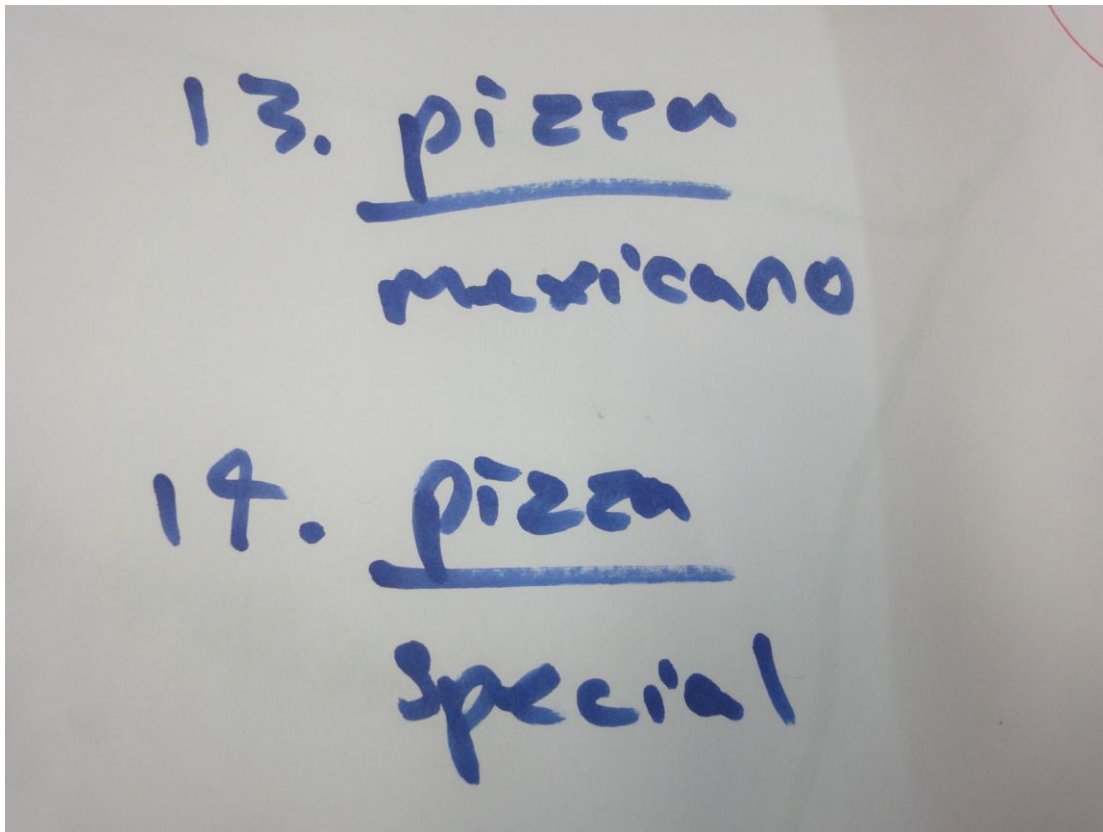
```

Result

```

pizza
-----
special
(1 row)

```



15. List pizzas, their country of origin and base for all pizzas containing fish ingredients.

```
select pizza, country, base
from menu
where pizza in (select pizza
from recipe
where ingredient in (select ingredient
from items
where type = 'fish'));
```

Result

pizza	country	base
-----+-----+-----		
napolitana	italy	Wf
stagiony	italy	Wm
special		Wf
siciliano	italy	Wm
seafood		Wm

(5 rows)

	Country	Base
Pizza		
Napolitana	Italy	Wf
Stagioni	Italy	Wm
Special		Wf
Siciliano	Italy	Wm
Seafood		Wm

16. Give the type of ingredients used for the most expensive pizza on the menu.

```
select distinct type
from items
where ingredient in (select ingredient
from recipe
where pizza in (select pizza
from menu
where price = (select max(price) from menu )));
```

Result

```
type
-----
dairy
fish
fruit
meat
spice
veg
(6 rows)
```