-- INNER JOIN

--list products with category name

-- select product\_id, product\_name, category\_id, category\_name

SELECT A.product\_id, A.product\_name, B.category\_id, B.category\_name

FROM production.products AS A

INNER JOIN production.categories AS B

ON A.category\_id = B.category\_id;

select product\_id, category\_name, product\_name

from production.categories

inner join production.products

on production.categories.category\_id = production.products.category\_id

-- list employees stores with their store information

-- select employee name, surname, store name

SELECT \*

from sales.staffs as a

inner join sales.stores as b

on a.store\_id = b.store\_id

SELECT sales.staffs.first\_name, sales.staffs.last\_name, sales.stores.store\_name

FROM sales.staffs

join sales.stores

on sales.staffs.store\_id = sales.stores.store\_id

SELECT ST.first\_name, ST.last\_name, SR.store\_name

FROM sales.staffs ST, sales.stores SR

WHERE ST.store\_id=SR.store\_id

-- LEFT JOIN

-- list products with category names

-- select product id, product name, category id, category names

select product\_id, category\_name, product\_name

from production.products

left join production.categories

on production.products.category\_id = production.categories.category\_id

SELECT TOP 20 A.product\_id, product\_name, B.category\_name

FROM production.products AS A

LEFT JOIN production.categories AS B

ON A.category\_id = B.category\_id

SELECT

A.product\_id,

A.product\_name,

B.category\_name

FROM production.products as A

LEFT JOIN production.categories B

ON A.category\_id = B.category\_id;

-- report the stock status of the products that product id greater than 310 in the store

-- expected columns: product id, product name, store id, quantity

SELECT A.product\_id, A.product\_name, B.store\_id, B.quantity

FROM production.products AS A

LEFT JOIN production.stocks AS B

ON A.product\_id = B.product\_id

WHERE A.product\_id > 310

SELECT

A.product\_id,

A.product\_name,

B.store\_id,

B.quantity

FROM production.products AS A

LEFT JOIN production.stocks AS B

ON A.product\_id = B.product\_id

WHERE A.product\_id > 310;

SELECT A.product\_id, A.product\_name, B.store\_id, B.quantity

FROM production.products A

LEFT JOIN production.stocks B

ON A.product\_id = B.product\_id

WHERE A.product\_id > 310

select a.product\_id, product\_name, store\_id, quantity from production.products as a

left join production.stocks as b

on a.product\_id = b.product\_id

where a.product\_id > 310

select A.product\_id,A.product\_name,B.store\_id, B.quantity

from production.products A

left join production.stocks B

on A.product\_id = B.product\_id

where A.product\_id > 310

-- RIGHT JOIN

-- report the stock status of the products that product id greater than 310 in the store

-- expected columns: product id, product name, store id, quantity

SELECT A.product\_id, A.product\_name, B.store\_id, B.quantity

FROM production.products AS A

RIGHT JOIN production.stocks AS B

ON A.product\_id = B.product\_id

WHERE A.product\_id > 310

SELECT

A.product\_id,

A.product\_name,

B.store\_id,

B.quantity

FROM production.stocks AS B

RIGHT JOIN production.products AS A

ON A.product\_id = B.product\_id

WHERE A.product\_id > 310;

select a.product\_id, product\_name, store\_id, quantity

from production.stocks as b

right join production.products as a

on a.product\_id = b.product\_id

where a.product\_id > 310

SELECT \*

FROM production.stocks B

RIGHT JOIN production.products A

ON A.category\_id = B.product\_id

WHERE A.product\_id > 310

select \*

from production.products A

RIGHT JOIN production.stocks B

ON A.product\_id = B.product\_id

Where A.product\_id > 310

SELECT \*

FROM production.stocks A

RIGHT JOIN production.products B

ON A.product\_id = B.product\_id

WHERE A.product\_id > 310

select A.product\_id,A.product\_name,B.store\_id, B.quantity

from production.stocks B

right join production.products A

on A.product\_id = B.product\_id

where A.product\_id > 310

SELECT A.product\_id, A.product\_name, B.store\_id, B.quantity

FROM production.stocks B

RIGHT JOIN production.products A

ON A.product\_id = B.product\_id

WHERE A.product\_id > 310

-- report the orders information made by all staffs

-- expected columns: staff\_id, first\_name, last\_name, all the information about orders

SELECT b.staff\_id, first\_name, last\_name

FROM sales.orders a

RIGHT JOIN sales.staffs b

ON a.staff\_id = b.staff\_id

SELECT \*

FROM sales.staffs A

RIGHT JOIN sales.orders B

ON A.staff\_id = B.staff\_id

select b.staff\_id, first\_name, last\_name, order\_status, order\_date, shipped\_date from sales.orders a

right join sales.staffs b

on b.staff\_id = a.staff\_id

SELECT A.staff\_id, A.first\_name,A.last\_name, B.\*

FROM sales.staffs A

RIGHT JOIN sales.orders B

ON A.staff\_id = B.staff\_id

SELECT A.first\_name, A.last\_name, A.staff\_id, B.\*

FROM sales.staffs A

RIGHT JOIN sales.orders AS B

ON A.staff\_id = B.staff\_id

SELECT

A.Staff\_id,

A.first\_name,

A.last\_name,

B.\*

FROM sales.orders AS B

RIGHT JOIN sales.staffs AS A

ON A.staff\_id = B.staff\_id;

SELECT A.staff\_id, A.first\_name, A.last\_name, B.\*

FROM sales.orders B

RIGHT JOIN sales.staffs A

ON A.staff\_id = B.staff\_id

-- FULL OUTER JOIN

-- write a query that returns stock and order information together for all products

-- expected columns: product\_id, store\_id, quantity, order\_id, list\_price

select \* from sales.orders A

full outer join production.stocks B

on a.store\_id =b.store\_id

select B.product\_id,B.store\_id,B.quantity,A.product\_id,A.list\_price,A.order\_id

from sales.order\_items A

full outer join production.stocks B

on A.product\_id=B.product\_id

select a.product\_id, B.store\_id, B.quantity,a.product\_id,a.list\_price, a.order\_id

from sales.order\_items a

full outer join production.stocks B

on a.product\_id = B.product\_id

order by a.product\_id

SELECT B.product\_id, B.store\_id, B.quantity, A.product\_id, A.list\_price, A.order\_id

FROM sales.order\_items A

FULL OUTER JOIN production.stocks B

ON A.product\_id = B.product\_id