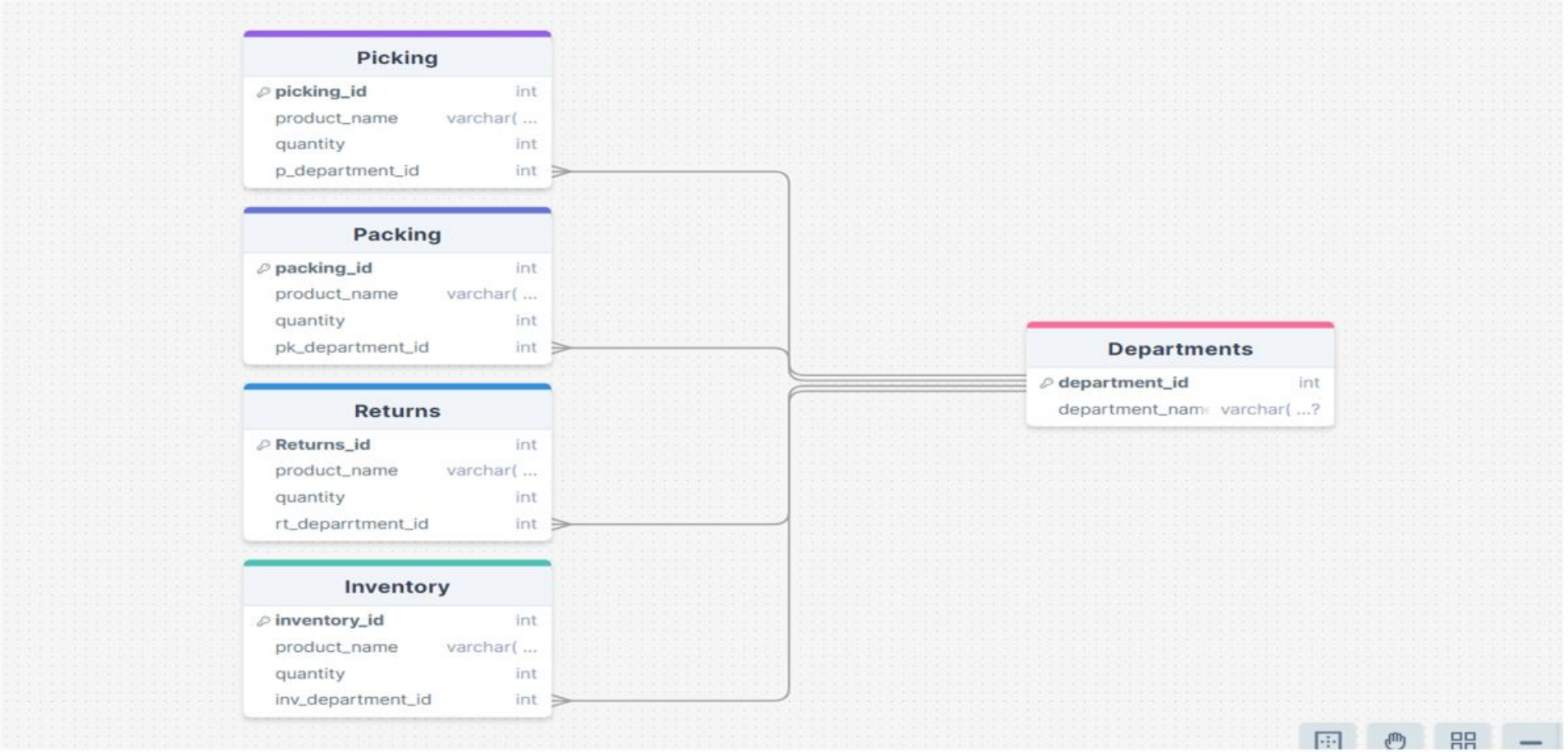
Managing Logistics Operations of

-GXO LOGISTICS-


GXO

ER Diagram for the schema's GXO Logistics



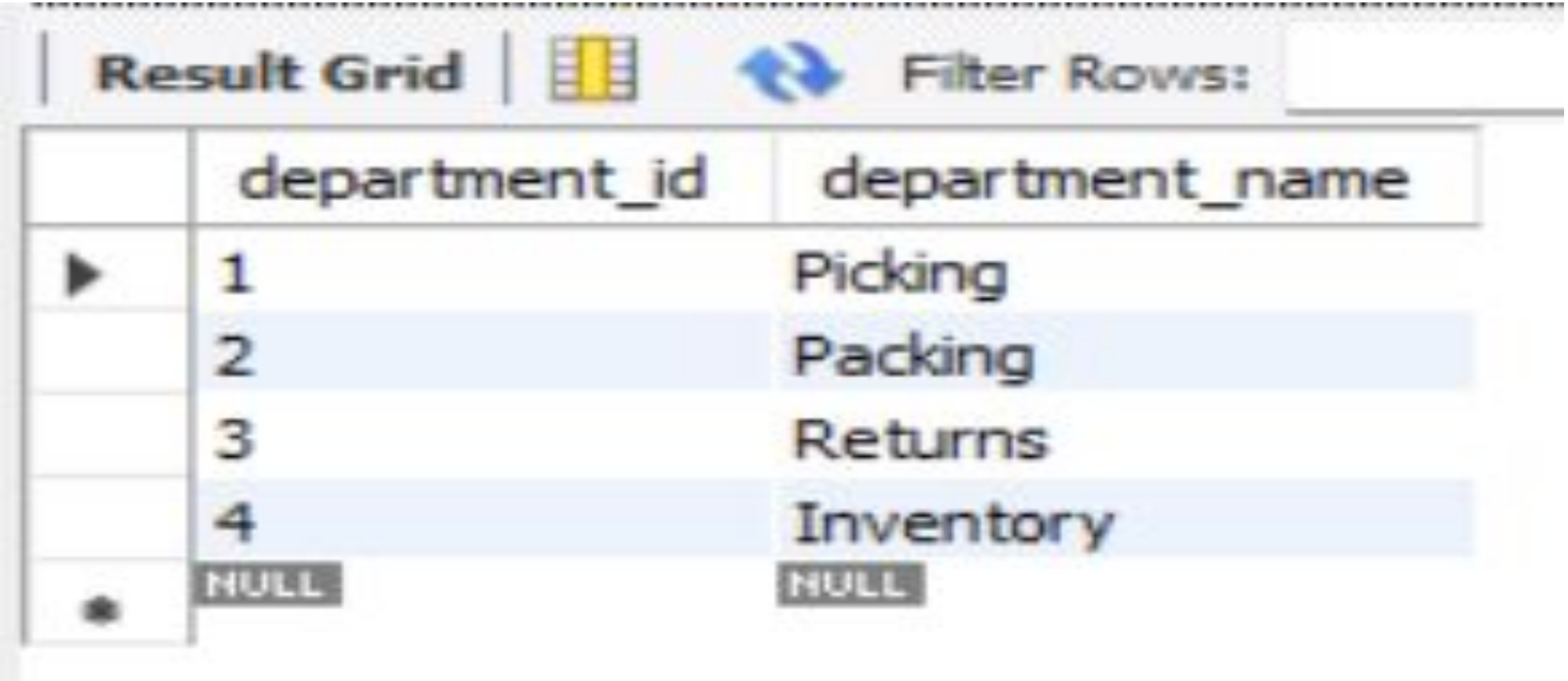
Structure of departments table

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
department_id	int(11)	NO	PRI	NULL	auto_increment
department_name	varchar(50)	NO		NULL	

```
4     -- Creating departments table
5 •   create table departments (
6         department_id INT auto_increment primary key,
7         department_name varchar(50) not null);
8
```

Content of department table using Select * from department;



	department_id	department_name
▶	1	Picking
	2	Packing
	3	Returns
	4	Inventory
*	NULL	NULL

Structure of picking table using desc picking;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
picking_id	int(11)	NO	PRI	NULL	auto_increment
product_name	varchar(100)	YES		NULL	
quantity	int(11)	YES		NULL	
p_department_id	int(11)	YES	MUL	NULL	

```
16 • create table picking (
17    picking_id int auto_increment primary key,
18    product_name varchar(100),
19    quantity int,
20    p_department_id int
21    );
22
23 • alter table picking add constraint fk_picking_department
24    foreign key (p_department_id) references departments(department_id);
25
```

Content of picking table using select * from picking;

Result Grid | Filter Rows: | Edit:

picking_id	product_name	quantity	p_department_id
1	Product A	10	1
2	Product B	20	1
3	Product C	30	1
4	Product D	40	1
5	Product E	50	1
6	Product F	60	1
7	Product G	70	1
8	Product H	80	1
9	Product I	90	1
10	Product J	100	1
11	Product K	110	1
12	Product L	120	1
13	Product M	130	1
14	Product N	140	1
15	Product O	150	1
16	Product P	160	1
17	Product Q	170	1
18	Product R	180	1
19	Product S	190	1
20	Product T	200	1

Structure of packing department table using Desc packing;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
packing_id	int(11)	NO	PRI	NULL	auto_increment
product_name	varchar(100)	YES		NULL	
quantity	int(11)	YES		NULL	
pk_department_id	int(11)	YES	MUL	NULL	

```
29 • create table packing (
30     packing_id int auto_increment primary key,
31     product_name varchar(100),
32     quantity int,
33     pk_department_id int
34     );
35
36 • alter table packing add constraint fk_packing_department
37     foreign key (pk_department_id) REFERENCES departments(department_id);
```

Content of packing department using select * from packing;

Result Grid | Filter Rows: | Edit:

packing_id	product_name	quantity	pk_department_id
1	Product A	20	2
2	Product B	30	2
3	Product C	40	2
4	Product D	50	2
5	Product E	60	2
6	Product F	70	2
7	Product G	80	2
8	Product H	90	2
9	Product I	100	2
10	Product J	110	2
11	Product K	120	2
12	Product L	130	2
13	Product M	140	2
14	Product N	150	2
15	Product O	160	2
16	Product P	170	2
17	Product Q	180	2
18	Product R	190	2
19	Product S	200	2
20	Product T	210	2

Structure of Returns table using Desc returns;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
return_id	int(11)	NO	PRI	NULL	auto_increment
product_name	varchar(100)	YES		NULL	
quantity	int(11)	YES		NULL	
rt_department_id	int(11)	YES	MUL	NULL	

```
41 • create table returns (
42     return_id int auto_increment primary key,
43     product_name varchar(100),
44     quantity int,
45     rt_department_id int
46     );
47
48 • alter table returns add constraint fk_returns_department
49     foreign key (rt_department_id) references departments(department_id);
```

Content of returns table using Select * from returns;

Result Grid | Filter Rows: | Edit:

return_id	product_name	quantity	rt_department_id
1	Product A	10	3
2	Product B	20	3
3	Product C	30	3
4	Product D	40	3
5	Product E	50	3
6	Product F	60	3
7	Product G	70	3
8	Product H	80	3
9	Product I	90	3
10	Product J	100	3
11	Product K	110	3
12	Product L	120	3
13	Product M	130	3
14	Product N	140	3
15	Product O	150	3
16	Product P	160	3
17	Product Q	170	3
18	Product R	180	3
19	Product S	190	3
20	Product T	200	3

Structure of inventory table using desc inventory;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
inventory_id	int(11)	NO	PRI	NULL	auto_increment
product_name	varchar(100)	YES		NULL	
quantity	int(11)	YES		NULL	
in_department_id	int(11)	YES	MUL	NULL	

```
54 • create table inventory (
55     inventory_id int auto_increment primary key,
56     product_name varchar(100),
57     quantity int,
58     in_department_id int
59     );
60
61 • alter table inventory add constraint fk_inventory_department
62     foreign key (in_department_id) references departments(department_id);
```

Content of inventory table using select * from inventory;

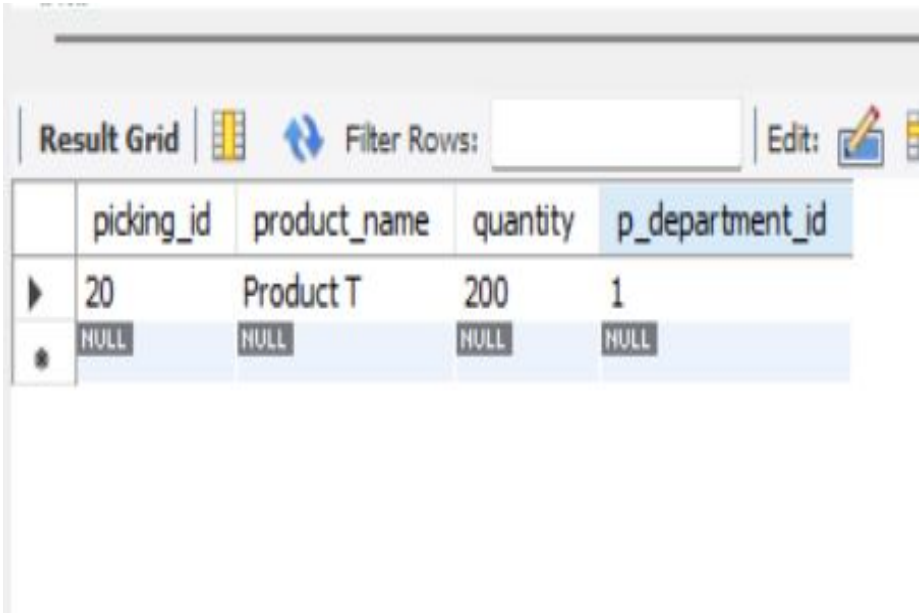
Result Grid | Filter Rows: | Edit:

inventory_id	product_name	quantity	in_department_id
1	Product A	10	4
2	Product B	20	4
3	Product C	30	4
4	Product D	40	4
5	Product E	50	4
6	Product F	60	4
7	Product G	70	4
8	Product H	80	4
9	Product I	90	4
10	Product J	100	4
11	Product K	110	4
12	Product L	120	4
13	Product M	130	4
14	Product N	140	4
15	Product O	150	4
16	Product P	160	4
17	Product Q	170	4
18	Product R	180	4
19	Product S	190	4
20	Product T	200	4

SUB-QUERIES

Retrieve the product with the maximum quantity in picking department

```
112    -- Retrieve the product with the maximum quantity in picking department
113
114 •  select * from picking
115     where quantity = (select max(quantity) from picking);
116
```



Result Grid | Filter Rows: | Edit:

picking_id	product_name	quantity	p_department_id
20	Product T	200	1
NULL	NULL	NULL	NULL

Finding the products in the returns department that have quantities matching those in the packing department

```
129
130 •  select product_name, quantity
131    from returns
132    where quantity in (SELECT quantity FROM packing);
133
```

Result Grid | Filter Rows:

product_name	quantity
Product D	40
Product E	50
Product F	60
Product G	70
Product H	80
Product I	90
Product J	100
Product K	110
Product L	120
Product M	130
Product N	140
Product O	150
Product P	160
Product Q	170
Product R	180
Product S	190
Product T	200

Update the quantities in the packing table to be 10 more than those in the inventory for the same product names

```
135
136 • update packing
137   set quantity = quantity + 10
138   where product_name in (select product_name from inventory);
139
```

Result Grid | Filter Rows: | Edit:

packing_id	product_name	quantity	pk_department_id
1	Product A	40	2
2	Product B	50	2
3	Product C	60	2
4	Product D	70	2
5	Product E	80	2
6	Product F	90	2
7	Product G	100	2
8	Product H	110	2
9	Product I	120	2
10	Product J	130	2
11	Product K	140	2
12	Product L	150	2
13	Product M	160	2
14	Product N	170	2
15	Product O	180	2
16	Product P	190	2
17	Product Q	200	2
18	Product R	210	2
19	Product S	220	2
20	Product T	230	2

JOINS

Get all products from picking along with their department names (LEFT JOIN)

```
144 •  select picking.product_name, picking.quantity, departments.department_name
145    from picking
146    left join departments
147    on p_department_id = departments.department_id;
```

Left join Output

Result Grid | Filter Rows:

product_name	quantity	department_name
Product A	10	Picking
Product B	20	Picking
Product C	30	Picking
Product D	40	Picking
Product E	50	Picking
Product F	60	Picking
Product G	70	Picking
Product H	80	Picking
Product I	90	Picking
Product J	100	Picking
Product K	110	Picking
Product L	120	Picking
Product M	130	Picking
Product N	140	Picking
Product O	150	Picking
Product P	160	Picking
Product Q	170	Picking
Product R	180	Picking
Product S	190	Picking

Get products and quantities that are present in both returns and packing (INNER JOIN)

```
158 •  select returns.product_name, returns.quantity, packing.quantity as packing_quantity
159    from returns
160    inner join packing
161    on returns.product_name = packing.product_name;
```

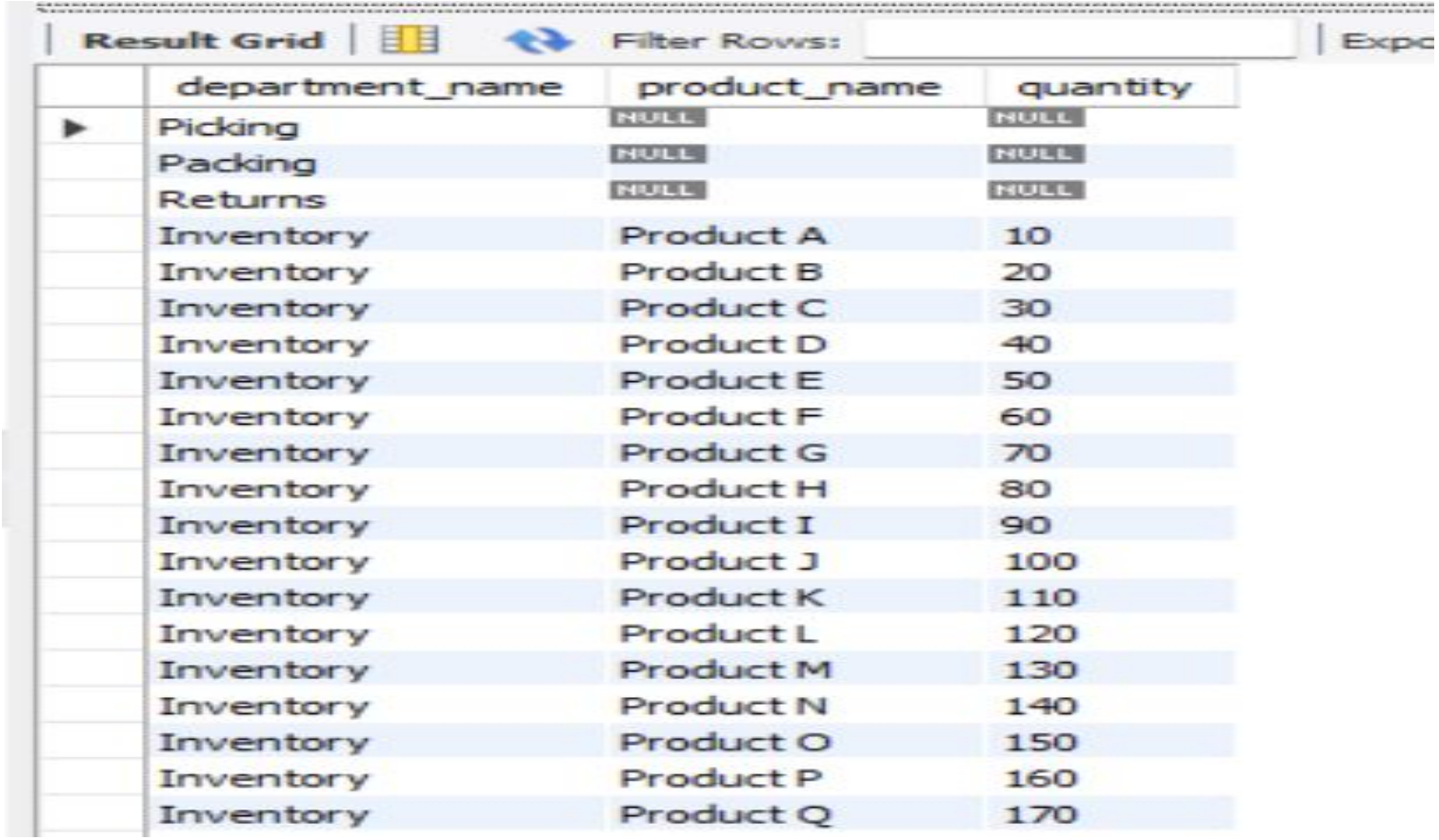
Inner Join Output

	product_name	quantity	packing_quantity
▶	Product A	10	40
	Product B	20	50
	Product C	30	60
	Product D	40	70
	Product E	50	80
	Product F	60	90
	Product G	70	100
	Product H	80	110
	Product I	90	120
	Product J	100	130
	Product K	110	140
	Product L	120	150
	Product M	130	160
	Product N	140	170
	Product O	150	180
	Product P	160	190
	Product Q	170	200
	Product R	180	210
	Product S	190	220
	Product T	200	230

Get all department names and their corresponding inventory products (RIGHT JOIN)

```
150
151 •  select departments.department_name, inventory.product_name, inventory.quantity
152    from inventory
153    right join departments
154    on in_department_id = departments.department_id;
```

Right Join Output



Result Grid | Filter Rows: | Expo

department_name	product_name	quantity
Picking	NULL	NULL
Packing	NULL	NULL
Returns	NULL	NULL
Inventory	Product A	10
Inventory	Product B	20
Inventory	Product C	30
Inventory	Product D	40
Inventory	Product E	50
Inventory	Product F	60
Inventory	Product G	70
Inventory	Product H	80
Inventory	Product I	90
Inventory	Product J	100
Inventory	Product K	110
Inventory	Product L	120
Inventory	Product M	130
Inventory	Product N	140
Inventory	Product O	150
Inventory	Product P	160
Inventory	Product Q	170

Get all pairs of products in the inventory table where the quantity of one is greater than the quantity of the other (SELF JOIN)

```
164
165 •   select a.product_name as Product1, b.product_name
166      as Product2, a.quantity
167      as Quantity1, b.quantity as Quantity2
168     from inventory a, inventory b
169     where a.quantity > b.quantity;
170
```

Self Join Output

Result Grid | Filter Rows:

Product1	Product2	Quantity1	Quantity2
Product N	Product M	140	130
Product O	Product M	150	130
Product P	Product M	160	130
Product Q	Product M	170	130
Product R	Product M	180	130
Product S	Product M	190	130
Product T	Product M	200	130
Product O	Product N	150	140
Product P	Product N	160	140
Product Q	Product N	170	140
Product R	Product N	180	140
Product S	Product N	190	140
Product T	Product N	200	140
Product P	Product O	160	150
Product Q	Product O	170	150
Product R	Product O	180	150
Product S	Product O	190	150
Product T	Product O	200	150
Product Q	Product P	170	160
Product R	Product P	180	160

Prepared by

Shubham Jagannath Bhosle

ITVedant

T318

DS/DA 3_5PM

THANK YOU