



Σύστημα Διαχείρισης Πόρων/Προμηθειών Βιομηχανίας

ΟΜΑΔΑ 6

Γεωργούλης Σταύρος 58388
Νίκου Τριαντάφυλλος-Μάριος 58487
Τουμαράς Γεώργιος 58420

Προσδιορισμός Απαιτήσεων

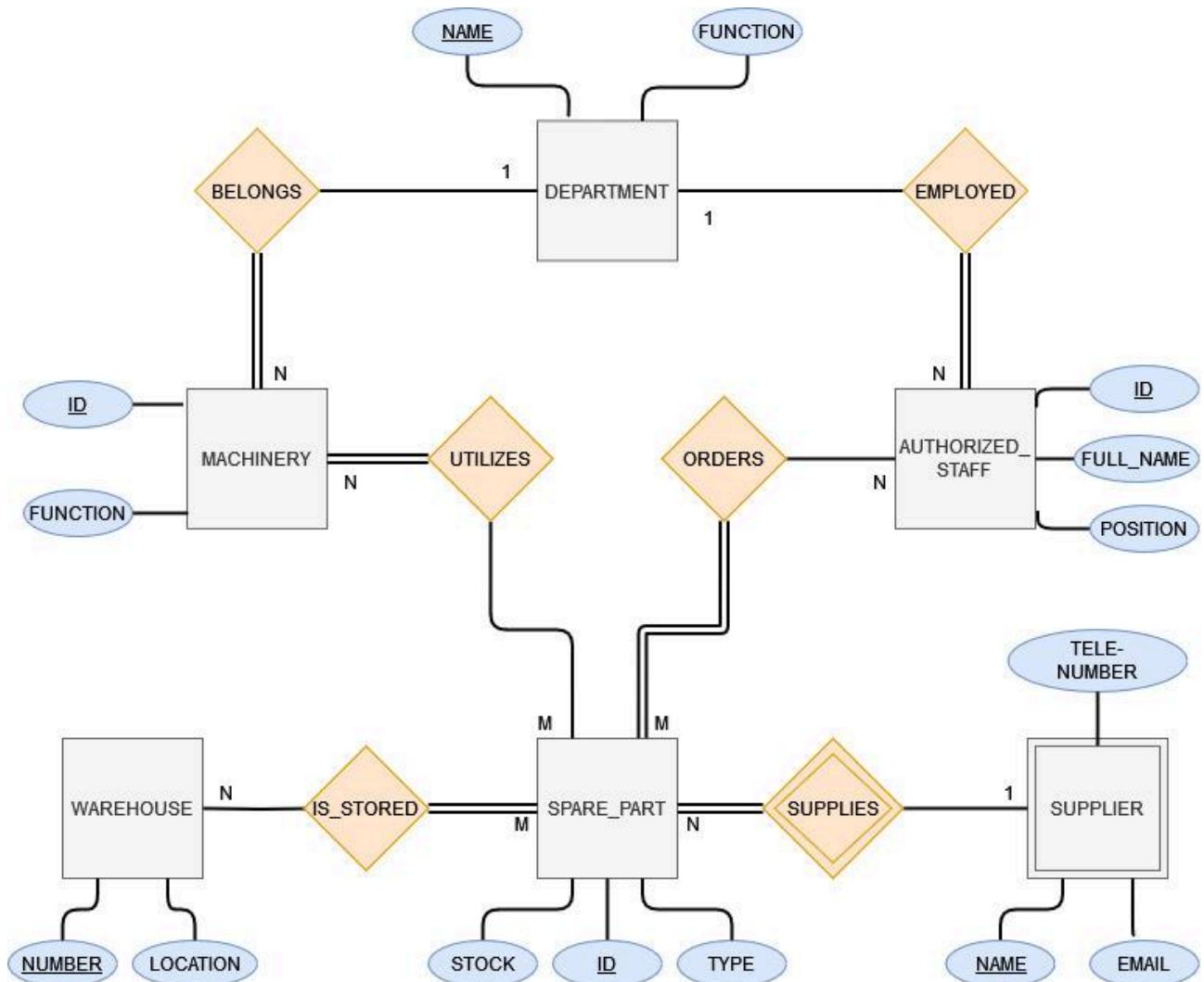
Η ομάδα μας κλήθηκε να δημιουργήσει μια βάση δεδομένων η οποία θα εξυπηρετεί τις ανάγκες ανεφοδιασμού μιας βιομηχανικής εγκατάστασης από τα αρμόδια άτομα. Για την εξακρίβωση των δυνατοτήτων της, ήρθαμε σε επαφή με τον πελάτη, ο οποίος έθεσε τις παρακάτω απαιτήσεις.

Απαιτήσεις

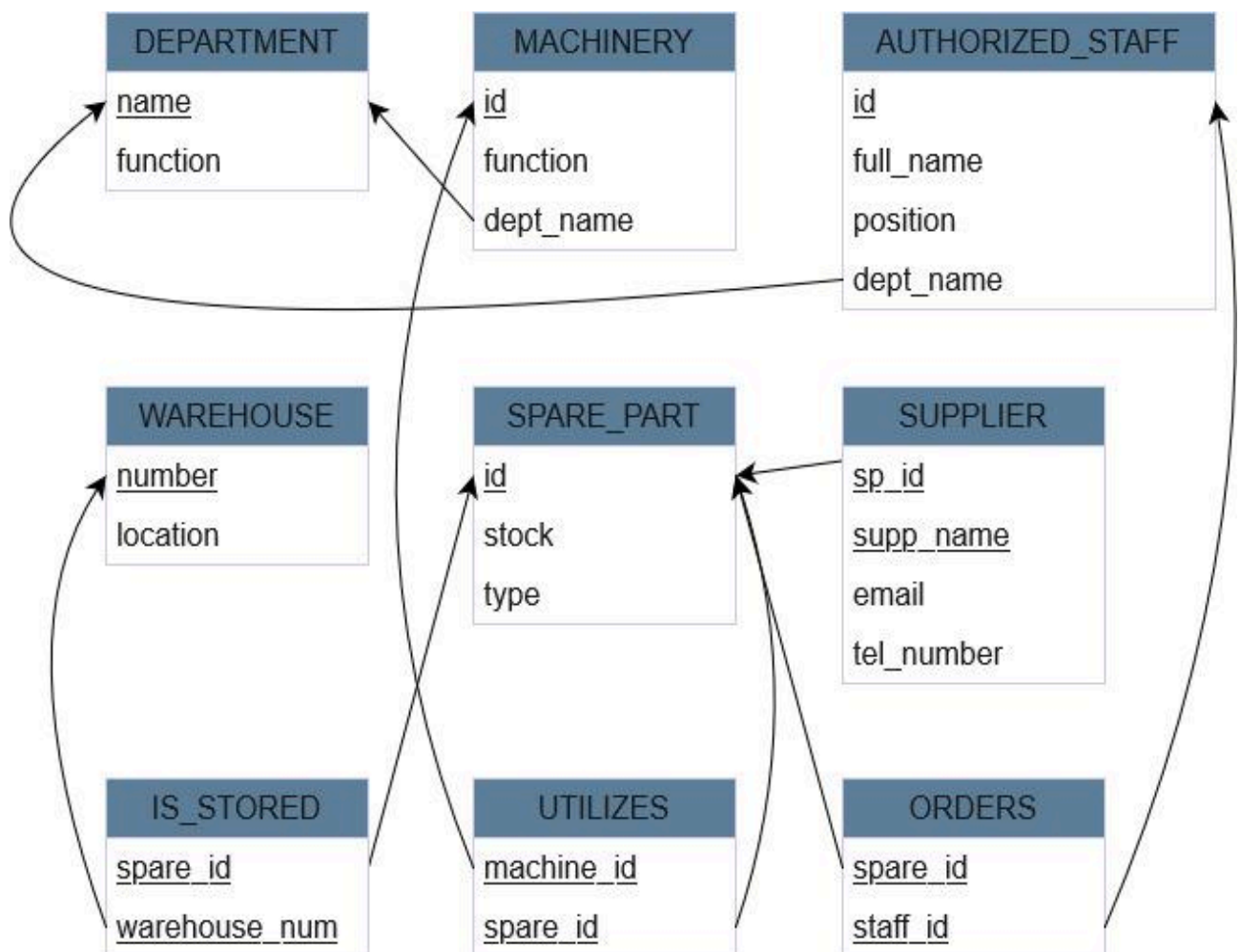
- Βασικό κομμάτι του εργοστασίου είναι τα μηχανήματα του, μέσω των οποίων γίνεται η παραγωγή.
- Κάθε μηχανήμα υπόκειται σε ένα μόνο τμήμα παραγωγής, έχει έναν κωδικό αναγνώρισης και επίσης σημειώνεται η λειτουργία που επιτελεί.
- Τα μηχανήματα αυτά χρησιμοποιούν ένα ή περισσότερα ανταλλακτικά, χαρακτηριστικά των οποίων είναι το είδος τους, ο μοναδικός κωδικός του καθενός και το απόθεμα τους.
- Τα τμήματα παραγωγής εκτός από τα μηχανήματα που τα αποτελούν, απασχολούν και προσωπικό .
- Το κάθε τμήμα παραγωγής έχει μια λειτουργία και ένα μοναδικό όνομα.
- Τα ανταλλακτικά αποθηκεύονται σε αποθήκες που για λόγους οργάνωσης είναι αριθμημένες, ενώ επίσης χρειάζεται να γνωρίζουμε την τοποθεσία τους.
- Κάθε είδος ανταλλακτικών αποθηκεύεται αποκλειστικά σε μία αποθήκη
- Τα ανταλλακτικά προμηθεύονται από προμηθευτές για τους οποίους πρέπει να γνωρίζουμε τα στοιχεία επικοινωνίας τους, δηλαδή το τηλέφωνο και την διεύθυνση email τους.
- Ανάλογα με τις ανάγκες κάθε τμήματος παραγωγής, το εξουσιοδοτημένο προσωπικό του τμήματος, έχει τη δυνατότητα να παραγγείλει ανταλλακτικά.
- Για το εξουσιοδοτημένο προσωπικό πρέπει να γνωρίζουμε τον αριθμό μητρώου του, το ονοματεπώνυμο του καθώς και την θέση του.

Διάγραμμα Οντοτήτων-Συσχετίσεων

Έχοντας αναλύσει τις απαιτήσεις μπορούμε πλέον να προχωρήσουμε στην σχεδίαση του διαγράμματος οντοτήτων-συσχετίσεων



Σχεσιακό Διάγραμμα



Εντολές SQL για τη δημιουργία της ΒΔ

Για την δημιουργία της βάσης δεδομένων και των πινάκων της αναπτύχθηκε ο παρακάτω κώδικας σε MySQL.

```
CREATE DATABASE supplytool;
```

```
CREATE TABLE authorized_staff(  
    id int(10) ,  
    full_name varchar(30) ,  
    position varchar(30) ,  
    dept_name varchar(20),  
    PRIMARY KEY (id),  
    FOREIGN KEY (dept_name) REFERENCES department(name)  
);
```

```
CREATE TABLE department(  
    name varchar(15) ,  
    function varchar(30),  
    PRIMARY KEY (name)  
);
```

```
CREATE TABLE is_stored (  
    spare_id int(10),  
    warehouse_num(11),  
    FOREIGN KEY(spare_id) REFERENCES spare_part(id),  
    FOREIGN KEY(warehouse_num) REFERENCES warehouse(number)  
);
```

```
CREATE TABLE machinery(  
    
```

```
id int(10) ,  
  
function varchar(30),  
  
dept_name varchar(20),  
  
PRIMARY KEY (id),  
  
FOREIGN KEY (dept_name) REFERENCES department(name) );
```

```
CREATE TABLE spare_part(  
  
id int(10) ,  
  
stock int(10),  
  
type varchar(20),  
  
PRIMARY KEY (id));
```

```
CREATE TABLE supplier (  
  
supp_name varchar(20),  
  
email varchar(30),  
  
tel_number int(10),  
  
spare_id int(10),  
  
PRIMARY KEY (supp_name),  
  
FOREIGN KEY (spare_id) REFERENCES spare_part(id)  
  
);
```

```
CREATE TABLE utilizes (  
  
spare_id int(10),  
  
machine_id int(10)  
  
FOREIGN KEY(spare_id) REFERENCES spare_part(id),  
  
  
FOREIGN KEY(machine_id) REFERENCES machinery(id)  
  
);
```

```
CREATE TABLE warehouse (  
    number int(10),  
    location varchar(20)  
    PRIMARY KEY (number)  
);
```

```
CREATE TABLE orders(  
    staff_id int(10),  
    spare_id int(10)  
    FOREIGN KEY(staff_id) REFERENCES authorized_staff(id),  
    FOREIGN KEY(spare_id) REFERENCES spare_part(id)  
);
```

Εντολές SQL για την εισαγωγή ενδεικτικών δεδομένων

//values for authorized_staff

```
insert into authorized_staff values(101,'Ioannidis Christos','Head_Enginner','Electronics'),  
insert into authorized_staff values(102,'Giannoulis Dimitris','Assistant_Enginner','Electronics'),  
insert into authorized_staff values(111,'Karakostas Anastasios','Head_Enginner','Production'),  
insert into authorized_staff values(112,'Anastasopoulos Petros','Assistant_Enginner','Production'),  
insert into authorized_staff values(121,'Stasinopoulos Christos','Head_Enginner','Maintenance'),  
insert into authorized_staff values(122,'Stathopoulou Georgia','Assistant_Enginner','Maintenance'),  
insert into authorized_staff values(131,'Leoudi Dimitra','Resource_Coordinator','Electronics'),
```

//values for department

```
insert into department values('Electronics','Elect. Maintenance');  
insert into department values('Logistics','Supply Planning');  
insert into department values('Maintenance','General Maintenance');  
insert into department values('Production','Product Manufacturing');
```

//values for machinery

insert into machinery values(361,'Welding Machine', 'Maintenance');

insert into machinery values(450,'P&P Machine', 'Electronics');

insert into machinery values(451,'PCB Etching', 'Electronics');

insert into machinery values(530,'Rolling Mill', 'Production');

insert into machinery values(531,'Painting Line', 'Production');

//values for spare_part

insert into spare_part values(4728, 7 , 'Motor(10kW)');

insert into spare_part values(4732, 10 , 'Motor(12kW)');

insert into spare_part values(4755, 250 , 'Fuse');

insert into spare_part values(4789, 130 , 'Relay');

insert into spare_part values(5049, 600 , 'Bearing');

insert into spare_part values(5052, 300 , 'Gear');

insert into spare_part values(5069, 10 , 'Roller');

insert into spare_part values(5077, 45 , 'Hydraulic Piston');

insert into spare_part values(5230, 700 , 'Lubrication Oil');

//values for warehouse

insert into warehouse values(1 , 'Sector A');

insert into warehouse values(2 , 'Sector c');

//values for supplier

insert into supplier values('ABB','Abb@supply.com' , 2105217638, 4728);

insert into supplier values('MECARESO', 'mecareso@supply.com',2116322541 , 5052);

insert into supplier values('honeywell', 'honeywell@supply.com',2101285279, 4755);

insert into supplier values('EFG Components','efg@supply.com' , 2101234567, 4732);

insert into supplier values('SKG Spare','sales@supply.gr' , 231011122, 4789);

insert into supplier values('SKG Spare Parts','sales@supply.gr' , 231011122, 5049);

insert into supplier values('Agrinio Supplies','agr@supply.gr' , 264111222, 5230);


```
insert into supplier values('FLR ','flr@supply.gr' , 231011342, 5069 );  
insert into supplier values('ELVAL','elvl@supply.gr' , 21071892, 5077 );
```

```
//values for is_stored
```

```
insert into is_stored VALUES (4789,1);  
insert into is_stored VALUES (4732,1);  
insert into is_stored VALUES (4728,1);  
insert into is_stored VALUES (5052,2);  
insert into is_stored VALUES (5049,2);  
insert into is_stored VALUES (5077,2);  
insert into is_stored VALUES (5069,1);  
insert into is_stored VALUES (5230,1);  
insert into is_stored VALUES (4755,1);
```

```
//values for orders
```

```
insert into orders VALUES (111,5069);  
insert into orders VALUES (102,4755);  
insert into orders VALUES (102,4789);  
insert into orders VALUES (131,5052);  
insert into orders VALUES (111,4728);  
insert into orders VALUES (111,5230);
```

```
//values for utilizes
```

```
insert into utilizes VALUES (5052,530);  
insert into utilizes VALUES (4755,451);  
insert into utilizes VALUES (4789,451);  
insert into utilizes VALUES (4732,531);  
insert into utilizes VALUES (5230,361);  
insert into utilizes VALUES (5069,361);
```

Ενδεικτικά ερωτήματα SQL

QUERIES:

- 1) Ποιοί είναι οι προμηθευτές και ποιά ανταλλακτικά παρέχουν;
- 2) Ποιά ανταλλακτικά έχω και σε ποιά αποθήκη βρίσκονται;
- 3) Πόσα άτομα βρίσκονται σε κάθε τμήμα;
- 4) Που βρίσκεται κάθε αποθήκη και ποιά ανταλλακτικά έχει η καθεμία;
- 5) Τι παραγγελίες έχει κάνει το κάθε προσωπικό και σε ποιο τμήμα ανήκει;
- 6) Ταξινόμησε μου αλφαβητικά όλα τα μέλη του προσωπικού

- 1) Αυτό το ερώτημα εμφανίζει τους προμηθευτές μαζί με τα ανταλλακτικά που παρέχουν.

```
SELECT sup.supp_name AS Supplier_Name, sup.email AS Supplier_Email, s.id AS Spare_Part_ID, s.type AS Spare_Part_Type
```

```
FROM supplier sup
```

```
JOIN spare_part s ON sup.spare_id = s.id;
```

Supplier_Name	Supplier_Email	Spare_Part_ID	Spare_Part_Type
ABB	Abb@supply.com	4728	Motor(10kW)
honeywell	honeywell@supply.com	4755	Fuse
MECARESO	mecareso@supply.com	5052	Gear

- 2) Αυτό το ερώτημα εμφανίζει τα ανταλλακτικά μαζί με τις αποθήκες στις οποίες είναι αποθηκευμένα.

```
SELECT s.id AS Spare_Part_ID, s.type AS Spare_Part_Type, w.location AS Warehouse_Location
```

```
FROM spare_part s
```

```
JOIN is_stored i ON s.id = i.spare_id
```

```
JOIN warehouse w ON i.warehouse_num = w.number;
```

Spare_Part_ID	Spare_Part_Type	Warehouse_Location
4789	Relay	Sector A
4732	Motor(12kW)	Sector A
4728	Motor(10kW)	Sector A
5052	Gear	Sector C
5049	Bearing	Sector C
5077	Hyrdaulic Piston	Sector C
5069	Roller	Sector A
5230	Lubricating Oil	Sector A
4755	Fuse	Sector A

3) Αυτό το ερώτημα εμφανίζει τα τμήματα μαζί με τον αριθμό των εξουσιοδοτημένων μελών προσωπικού σε κάθε τμήμα.

SELECT d.name AS Department_Name, COUNT(a.id) AS Authorized_Staff_Count

FROM department d

LEFT JOIN authorized_staff a **ON** d.name = a.dept_name

GROUP BY d.name;

Department_Name	Authorized_Staff_Count
Electronics	2
Logistics	1
Maintenance	2
Production	2

4) Αυτό το ερώτημα εμφανίζει τις τοποθεσίες των αποθηκών μαζί με τον αριθμό των ανταλλακτικών που αποθηκεύονται σε κάθε αποθήκη.

```
SELECT w.location AS Warehouse_Location, COUNT(i.spare_id) AS Stored_Spare_Parts_Count
FROM warehouse w
LEFT JOIN is_stored i ON w.number = i.warehouse_num
GROUP BY w.location;
```

Warehouse_Location	Stored_Spare_Parts_Count
Sector A	6
Sector C	3

5) Αυτό το ερώτημα εμφανίζει τα μέλη του προσωπικού, τα τμήματά τους και τις παραγγελίες που έχουν κάνει αυτά τα μέλη του προσωπικού.

```
SELECT a.full_name AS Staff_Name, d.name AS Department_Name, o.spare_id AS Ordered_Spare_Part
FROM authorized_staff a
JOIN department d ON a.dept_name = d.name
LEFT JOIN orders o ON a.id = o.staff_id;
```

Staff_Name	Department_Name	Ordered_Spare_Part
Karakostas Anastasios	Production	5069
Giannoulis Dimitris	Electronics	4755
Giannoulis Dimitris	Electronics	4789
Leoudi Dimitra	Logistics	5052
Karakostas Anastasios	Production	4728
Karakostas Anastasios	Production	5230
Ioannidis Christos	Electronics	NULL
Anastsopoulos Petros	Production	NULL
Stasinopoulos Christos	Maintenance	NULL
Stathopoulou Georgia	Maintenance	NULL

6) Αυτό το ερώτημα ανακτά τα μέλη του εξουσιοδοτημένου προσωπικού και τα ταξινομεί αλφαβητικά με βάση τα πλήρη ονόματά τους.

```
SELECT full_name, position
```

```
FROM authorized_staff
```

```
ORDER BY full_name ASC;
```

full_name	position
Anastsopoulos Petros	Assistant_Enginner
Giannoulis Dimitris	Assistant_Enginner
Ioannidis Christos	Head_Enginner
Karakostas Anastasios	Head_Enginner
Leoudi Dimitra	Resource Coordinator
Stasinopoulos Christos	Head_Enginner
Stathopoulou Georgia	Assistant_Enginner

7 rows in set (0.00 sec)

SQL QUERIES

ΕΡΩΤΗΜΑ 1

a. 1.(Inner Join)

Q:Για ποιά ανταλλακτικά έχω προμηθευτή;

SELECT spare_part.type, supplier.supp_name

FROM spare_part

INNER JOIN supplier **ON** spare_part.id = supplier.spare_id;

```
mysql> SELECT spare_part.type, supplier.supp_name
-> FROM spare_part
-> INNER JOIN supplier ON spare_part.id = supplier.spare_id;
+-----+-----+
| type      | supp_name |
+-----+-----+
| Motor(10kW) | ABB       |
| Fuse        | honeywell |
| Gear        | MECARESO  |
+-----+-----+
3 rows in set (0.00 sec)
```

Αυτό το ερώτημα εμφανίζει μόνο τα ανταλλακτικά που έχουν προμηθευτή.Εφόσον το 'Motor(10kW)' δεν έχει προμηθευτή δεν θα εμφανιστεί.

2.(Left Outer Join)

Q:Δείξε μου όλα τα ανταλλακτικά και σε ποιά έχω προμηθευτή;

SELECT spare_part.type, supplier.supp_name

FROM spare_part

LEFT OUTER JOIN supplier **ON** spare_part.id = supplier.spare_id;

```
mysql> SELECT spare_part.type, supplier.supp_name
-> FROM spare_part
-> LEFT OUTER JOIN supplier ON spare_part.id = supplier.spare_id;
+-----+-----+
| type          | supp_name |
+-----+-----+
| Motor(10kW)   | ABB       |
| Motor(12kW)   | NULL      |
| Fuse          | honeywell |
| Relay         | NULL      |
| Bearing       | NULL      |
| Gear          | MECARESO  |
| Roller        | NULL      |
| Hyrdaulic Piston | NULL      |
| Lubricating Oil | NULL      |
+-----+-----+
9 rows in set (0.00 sec)
```

Αυτό το ερώτημα θα περιλαμβάνει όλα τα ανταλλακτικά, ακόμη και αυτά που δεν έχουν προμηθευτή. Το 'Motor(10kW)' (ID: 4728) θα εμφανιστεί στα αποτελέσματα με μια τιμή NULL στη στήλη supp_name επειδή δεν έχει προμηθευτή.

b. 1. Q:Δείξε μου τον αριθμό των παραγγελιών για κάθε ανταλλακτικό;

```
SELECT spare_id, COUNT(*) AS num_orders
```

```
FROM orders
```

```
GROUP BY spare_id;
```

```
mysql> SELECT spare_id, COUNT(*) AS num_orders
-> FROM orders
-> GROUP BY spare_id;
+-----+-----+
| spare_id | num_orders |
+-----+-----+
| 4728     | 1         |
| 4755     | 1         |
| 4789     | 1         |
| 5052     | 1         |
| 5069     | 1         |
| 5230     | 1         |
+-----+-----+
6 rows in set (0.00 sec)
```

Αυτή η εντολή χρησιμοποιεί τη συνάρτηση COUNT(*) για να μετρήσει πόσες παραγγελίες έχουν γίνει για κάθε ανταλλακτικό.

2.

Q:Δείξε μου το συνολικό απόθεμα για όλα τα ανταλλακτικά;

```
SELECT SUM(stock) AS total_stock
```

```
FROM spare_part;
```

```
mysql> SELECT SUM(stock) AS total_stock
-> FROM spare_part;
+-----+
| total_stock |
+-----+
|          2052 |
+-----+
1 row in set (0.01 sec)
```

Αυτή η εντολή χρησιμοποιεί τη συνάρτηση **SUM(stock)** για να υπολογίσει το συνολικό απόθεμα όλων των ανταλλακτικών.

c.1.

Q:Ποιά ανταλλακτικά έχω σε απόθεμα με ποσότητα μεγαλύτερη του 100;

```
SELECT id, SUM(stock) AS total_quantity
```

```
FROM spare_part
```

```
GROUP BY id
```

```
HAVING SUM(stock) > 100;
```

```
mysql> SELECT id, SUM(stock) AS total_quantity
-> FROM spare_part
-> GROUP BY id
-> HAVING SUM(stock) > 100;
+-----+-----+
| id    | total_quantity |
+-----+-----+
| 4755  |          250  |
| 4789  |          130  |
| 5049  |          600  |
| 5052  |          300  |
| 5230  |          700  |
+-----+-----+
5 rows in set (0.00 sec)
```

Αυτό το ερώτημα προσθέτει το πεδίο 'stock' για κάθε μοναδικό 'id' στον πίνακα 'spare_part' και επιστρέφει τη συνολική ποσότητα για κάθε ανταλλακτικό, φιλτράροντας αυτά με συνολική ποσότητα μεγαλύτερη των 100.

2.

Q: Ποιά τμήματα έχουν περισσότερα από ένα μηχανήματα;

SELECT dept_name, COUNT(*) AS num_machines

FROM machinery

GROUP BY dept_name

HAVING COUNT(*) > 1;

```
mysql> SELECT dept_name, COUNT(*) AS num_machines
-> FROM machinery
-> GROUP BY dept_name
-> HAVING COUNT(*) > 1;
+-----+-----+
| dept_name | num_machines |
+-----+-----+
| Electronics | 2 |
| Production | 2 |
+-----+-----+
2 rows in set (0.01 sec)
```

Αυτό το ερώτημα ομαδοποιήσει τα μηχανήματα ανά τμήμα και επιστρέψει τον αριθμό των μηχανημάτων, αποκλείοντας τμήματα που έχουν λιγότερα από 2 μηχανήματα.

ΕΡΩΤΗΜΑ 2

α) Όταν εξετάζουμε τις συναρτήσεις παρατηρούμε τα εξής :

- 1) το γνώρισμα E εμφανίζεται στις εξαρτήσεις 2 και 4
- 2) το γνώρισμα A εμφανίζεται στις εξαρτήσεις 1 και 2
- 3) κανένα άλλο γνώρισμα δεν εμφανίζεται σε όλες τις συναρτήσεις

Αυτο σημαίνει ότι το σύνολο {E,A} μπορεί να χρησιμοποιηθεί ως κλειδί, διότι καθορίζει μοναδικά κάθε εγγραφή στη βάση δεδομένων .

b) Το σχήμα σε 2nf είναι το εξής :

- 1) $R_1(\{A,J\})$: περιλαμβάνει το πρωτεύον κλειδί {A} και το γνώρισμα J που εξαρτάται από αυτό
- 2) $R_2(\{E,A,F\})$: περιλαμβάνει το πρωτεύον κλειδί {E,A} και το γνώρισμα F που εξαρτάται από αυτό
- 3) $R_3(\{E,G,D\})$: περιλαμβάνει το πρωτεύον κλειδί {E,A} και τα μη κλειδωμένα γνωρίσματα G,D που εξαρτώνται από αυτό
- 4) $R_4(\{E,A,H\})$: περιλαμβάνει το πρωτεύον κλειδί {E,A} και το γνώρισμα H που εξαρτάται από αυτό
- 5) $R_5(\{A,J,C,B\})$: περιλαμβάνει το πρωτεύον κλειδί {A} και τα μη κλειδωμένα γνωρίσματα J,C,B που εξαρτώνται από αυτό

c) το σχήμα σε 3nf είναι το εξής

- 1) $R_1(\{A,J\})$: περιλαμβάνει το πρωτεύον κλειδί {A} και το γνώρισμα J που εξαρτάται από αυτό
- 2) $R_2(\{E,A,F\})$: περιλαμβάνει το πρωτεύον κλειδί {E,A} και το γνώρισμα F που εξαρτάται από αυτό
- 3) $R_{3a}(\{E,G\})$
- 4) $R_{3b}(\{e,D\})$
- 5) $R_4(\{E,A,H\})$: περιλαμβάνει το πρωτεύον κλειδί {E,A} και το γνώρισμα H που εξαρτάται από αυτό
- 6) $R_5(\{A,J,C,B\})$: περιλαμβάνει το πρωτεύον κλειδί {A} και τα μη κλειδωμένα γνωρίσματα J,C,B που εξαρτώνται από αυτό

ΕΡΩΤΗΜΑ 3

A) Οι δύο μεγαλύτεροι πίνακες της βάσης μας είναι οι πίνακες supplier και authorized staff διότι έχουν και τα περισσότερα γνωρίσματα .Θα ερευνήσουμε τον κάθε πίνακα ξεχωριστά.

Πίνακας supplier

Ο πίνακας supplier αποτελείται από τα εξής γνωρίσματα :

- 1) Sp_id
- 2) Supp_name
- 3) Email
- 4) Tel_number

Οι συναρτησιακές εξαρτήσεις από τον πίνακα supplier προκύπτουν οι εξής:

Supp_name -> email, tel_number, spare_id

Spare_id->spare_part

Από τις συναρτησιακές εξαρτήσεις προκύπτει ότι το 'supp_name' ορίζει ολοκληρωτικά τα 'email', 'tel_number' και 'spare_id', οπότε είναι το πρωτεύον κλειδί. Παρατηρούμε ότι πράγματι συμπίπτει το κλειδί από τις συναρτησιακές εξαρτήσεις με το κλειδί που είχαμε ορίσει εμείς.

Πίνακας authorized staff

Ο πίνακας authorized staff αποτελείται από τα εξής γνωρίσματα:

- 1) Id
- 2) Full_name
- 3) Position
- 4) Dept_name

Οι συναρτησιακές εξαρτήσεις από τον πίνακα authorized staff προκύπτουν οι εξής:

Id->full_name,position,dept_name

Dept_name->department

Από τις συναρτησιακές σχέσεις προκύπτει ότι το 'id' ορίζει ολοκληρωτικά τα 'full_name', 'position', 'dept_name', οπότε είναι και το πρωτεύον κλειδί. Συγκρίνοντας με το πρωτεύον κλειδί που είχαμε ορίσει εμείς, παρατηρούμε ότι συμπίπτει, οπότε η χρήση του 'id' για πρωτεύον κλειδί θεωρείται σωστή.

B) Η 2NF απαιτεί κάθε μη-κλειδικό γνώρισμα να εξαρτάται είτε από το πρωτεύον κλειδί είτε από κάποιο άλλο μη-κλειδικό γνώρισμα. Εάν κάποιο μη-κλειδικό γνώρισμα εξαρτάται μόνο από μερικά κομμάτια του πρωτεύοντος κλειδιού, η σχέση δεν είναι σε 2NF.

Για τον πίνακα "supplier", το 'supp_name' είναι το πρωτεύον κλειδί. Εξαρτάται από το 'supp_name', ενώ το 'spare_id' εξαρτάται από το πρωτεύον κλειδί. Καθώς και τα 'email', 'tel_number' εξαρτώνται μόνο από το 'supp_name', ο πίνακας είναι σε 2NF.

Για τον πίνακα authorized staff το id είναι το πρωτεύον κλειδί και όλα τα μη κλειδικά γνωρίσματα εξαρτώνται ολοκληρωτικά από αυτό άρα είναι και αυτός σε 2NF.

Γ) Η 3NF απαιτεί κάθε μη-κλειδικό γνώρισμα να εξαρτάται είτε από το πρωτεύον κλειδί είτε από κάποιο άλλο μη-κλειδικό γνώρισμα. Εάν κάποιο μη-κλειδικό γνώρισμα εξαρτάται αποκλειστικά από άλλο μη-κλειδικό γνώρισμα, τότε η σχέση δεν είναι σε 3NF.

Όπως και στον πίνακα "supplier" και στο "authorized staff", κανένα μη-κλειδικό γνώρισμα δεν εξαρτάται αποκλειστικά από άλλο μη-κλειδικό γνώρισμα, οπότε είναι σε 3NF μορφή.

ΕΡΩΤΗΜΑ 4 (SQL COMMANDS)

a. Μια δοσοληψία (transaction) με παραδείγματα εντολών commit και rollback

```
START TRANSACTION;
```

```
// Ελέγχουμε αν η απόθεση του ανταλλακτικού είναι αρκετή//
```

```
IF (SELECT stock FROM spare_part WHERE id = 123) >= 5 THEN
```

```
    // Αν η απόθεση είναι αρκετή, μειώνουμε το stock και κάνουμε COMMIT//
```

```
    UPDATE spare_part SET stock = stock - 5 WHERE id = 123;
```

```
    COMMIT;
```

```
ELSE
```

```
    // Αν δεν είναι αρκετή, κάνουμε ROLLBACK//
```

```
    ROLLBACK;
```

```
END IF;
```

1. **START TRANSACTION:** Ξεκινά μια συναλλαγή, επιτρέποντας πολλαπλές ενέργειες να θεωρούνται ως μια μοναδική, ατομική αλλαγή.
2. **Stock Check:** Με χρήση IF, ελέγχει αν η ποσότητα αποθέματος ενός συγκεκριμένου ανταλλακτικού (με αναγνωριστικό 123) είναι μεγαλύτερη ή ίση με 5.
3. **Stock Update:** Αν το απόθεμα είναι επαρκές (5 ή περισσότερο), εκτελείται μια εντολή UPDATE για να μειωθεί το απόθεμα του ανταλλακτικού κατά 5.
4. **COMMIT:** Εάν τα δεδομένα ενημερώθηκαν με επιτυχία και το απόθεμα μειώθηκε, εφαρμόζεται η εντολή COMMIT, ολοκληρώνοντας τις αλλαγές στη βάση δεδομένων.
5. **ROLLBACK:** Εάν το απόθεμα δεν είναι επαρκές, εκτελείται η εντολή ROLLBACK, αναιρώντας οποιοδήποτε αλλαγές που έγιναν μέσα στη συναλλαγή και διατηρώντας τη συνέπεια των δεδομένων.

b. Ευρετήρια σε δύο πίνακες της επιλογής σας. Να δικαιολογήσετε την επιλογή των πινάκων/γνωρισμάτων.

Η δημιουργία ευρετηρίων εξυπηρετεί την ταχύτητα αναζήτησης σε μεγάλους πίνακες. Ένας καλός υποψήφιος για ευρετήριο είναι ένας πίνακας που συχνά χρησιμοποιείται σε ερωτήματα αναζήτησης ή συγχώνευσης. Γενικά, τα πεδία που χρησιμοποιούνται για συχνές αναζητήσεις είναι καλοί υποψήφιοι για ευρετήριο.

Ένα παράδειγμα δημιουργίας ευρετηρίου σε έναν πίνακα orders για τα πεδία staff_id και spare_id μπορεί να είναι:

```
CREATE INDEX idx_staff_id ON orders(staff_id);  
CREATE INDEX idx_spare_id ON orders(spare_id);
```

Σε αυτό το παράδειγμα, δημιουργούμε ευρετήρια για τα πεδία staff_id και spare_id του πίνακα orders, καθώς χρησιμοποιούνται συχνά σε ερωτήματα αναζήτησης ή συνδυασμού με άλλους πίνακες.

c. Δύο Όψεις (με απλά αλλά και σύνθετα ερωτήματα).

Όψη 1:

Αυτό το view θα δείχνει λεπτομέρειες σχετικά με τις παραγγελίες, το υπόλοιπο απόθεμα των ανταλλακτικών και τις αντίστοιχες λεπτομέρειες του προμηθευτή για αυτά τα ανταλλακτικά.

```
CREATE VIEW OrderDetails AS  
  
SELECT o.staff_id, o.spare_id, s.stock, s.type AS spare_part,  
       su.supp_name AS supplier_name, su.email AS supplier_email, su.tel_number AS supplier_tel  
  
FROM orders o  
  
JOIN spare_part s ON o.spare_id = s.id  
  
JOIN supplier su ON s.id = su.spare_id;
```

Όψη 2:

Ας υποθέσουμε ότι θέλουμε μια όψη που δείχνει το προσωπικό, τα τμήματα στα οποία ανήκουν, τα μηχανήματα που χρησιμοποιούν τα τμήματα και τα ανταλλακτικά που σχετίζονται με αυτό το μηχάνημα.

```
CREATE VIEW StaffDepartmentMachinery AS  
  
SELECT a.full_name AS Employee_Name, a.position AS Position, b.name AS Department,  
       c.function AS Machinery_Function, d.type AS Spare_Part_Type  
  
FROM authorized_staff a  
  
JOIN department b ON a.dept_name = b.name
```

```
LEFT JOIN machinery c ON a.dept_name = c.dept_name
```

```
LEFT JOIN utilizes e ON c.id = e.machine_id
```

```
LEFT JOIN spare_part d ON e.spare_id = d.id;
```

d. Μία Stored Procedure και μία Stored Function.

Stored Function

```
DELIMITER //
```

```
CREATE FUNCTION GetTotalStock(sparePartID INT) RETURNS INT
```

```
BEGIN
```

```
    DECLARE totalStock INT;
```

```
    SELECT SUM(stock) INTO totalStock
```

```
    FROM spare_part
```

```
    WHERE id = sparePartID;
```

```
    RETURN totalStock;
```

```
END //
```

```
DELIMITER ;
```

Αυτή η αποθηκευμένη συνάρτηση επιστρέφει το συνολικό απόθεμα ενός συγκεκριμένου ανταλλακτικού, λαμβάνοντας υπόψη το ID του ανταλλακτικού.

-- Call the function--

```
SELECT GetTotalStock(4732); -- Replace '4732' with the spare part ID you want to retrieve the total stock for
```

Stored Procedure

DELIMITER //

```
CREATE PROCEDURE TotalSPInWarehouses()
BEGIN
    SELECT warehouse_num, COUNT(spare_id) AS Total_Spare_Parts
    FROM is_stored
    GROUP BY warehouse_num;
END //
```

DELIMITER ;

Αυτή η αποθηκευμένη διαδικασία επιστρέφει τον αριθμό των ανταλλακτικών ανά αποθήκη.

-- Call the procedure--

```
CALL TotalSPInWarehouses();
```

e. Ένα Trigger.

// Δημιουργία νέου πίνακα//

```
CREATE TABLE notification_queue (
    id INT AUTO_INCREMENT PRIMARY KEY,
    message VARCHAR(255),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
ALTER TABLE notification_queue
```

```
ADD COLUMN spare_part_id INT,
```

```
ADD FOREIGN KEY (spare_part_id) REFERENCES spare_part(id);
```

// Δημιουργία νέου πίνακα//

DELIMITER //

CREATE TRIGGER LowStockNotification

AFTER UPDATE ON spare_part

FOR EACH ROW

BEGIN

IF NEW.stock < 50 THEN

INSERT INTO notification_queue (message) VALUES (CONCAT('Low stock for part ID: ', CAST(NEW.id AS CHAR)));

END IF;

END

//

DELIMITER ;

Αυτός ο κώδικας δημιουργεί ένα trigger στον πίνακα spare_part. Το trigger αυτό εκτελείται κάθε φορά που γίνεται μια ενημέρωση (UPDATE) στον πίνακα spare_part. Συγκεκριμένα, ελέγχει αν η νέα τιμή του stock (απόθεμα) είναι μικρότερη από 50. Αν ισχύει αυτό, τότε εισάγει ένα μήνυμα ειδοποίησης στον πίνακα notification_queue, το οποίο περιέχει το μήνυμα "Χαμηλό απόθεμα για το είδος με αναγνωριστικό: [ID του αντικειμένου]".

```
mysql> select * from notification_queue;
+----+-----+-----+
| id | message                                | created_at |
+----+-----+-----+
| 1  | Low stock for part ID: 5077 | 2024-01-03 16:18:28 |
+----+-----+-----+
1 row in set (0.00 sec)
```

ΕΡΩΤΗΜΑ 5

Η Web εφαρμογή Supply Tool που λειτουργεί σε αλληλεπίδραση με την βάση δεδομένων μας μπορεί να βρεθεί εδώ: <https://dblab.nonrelevant.net/~lab2324omada6/>