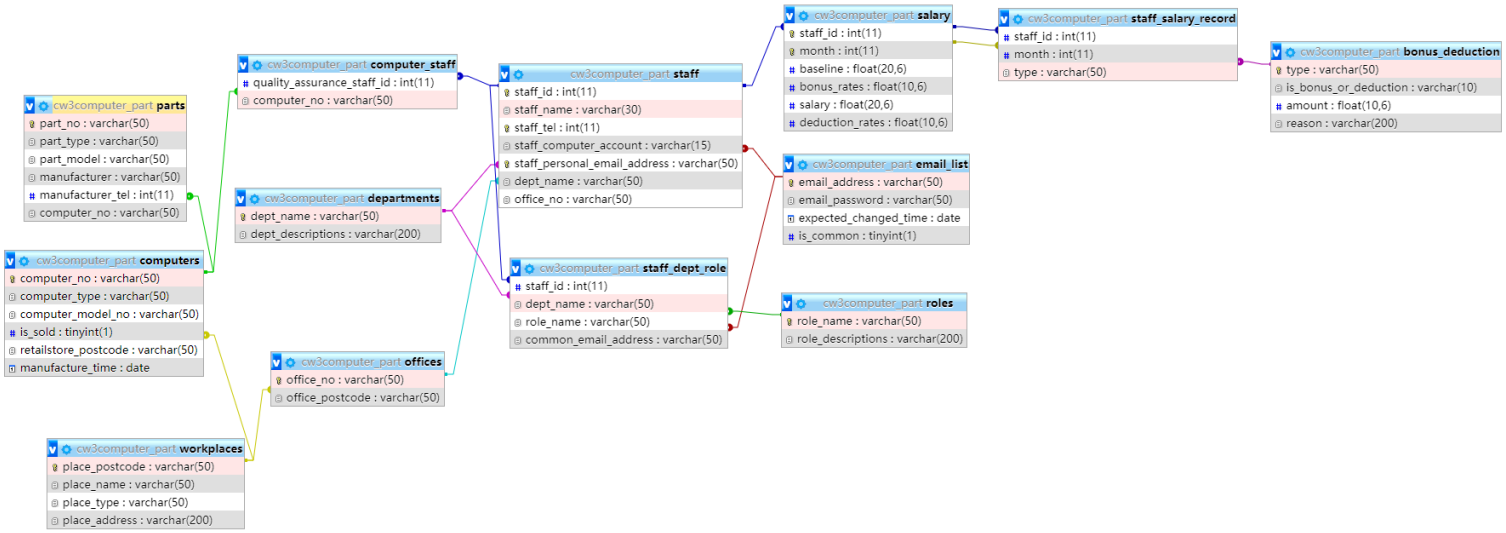
Report for CW3 Design

ER Diagram



1. M:M relationship:

1) The relationship between **staff** and **roles**
   Reason: “Each staff is assigned to one or more roles in his department. Two or more persons may share the same role in the department”
2) The relationship between **dept** and **roles**
   Reason: “Two departments may have roles with the same name”, and every department has different roles.
3) The relationship between **salary** and **bonus_deduction**
   Reason: The salary of every staff is relevant with various bonus and deduction types.
   At the same time, every bonus and deduction type can influence the salaries of different staff.
4) The relationship between **staff** and **computers**
   Reason: computers can be examined by various staff, while every staff can examine many computers.

To solve these five M:M relationships, we could create an additional entity to transform M:M to two M:1, or transform two M:M into three M:1.

So, I create 4 new tables.

1) staff_dept_role: {staff_id, dept_name, role_name, common_email_address}
   - **components**
     - staff_id is PK in the “staff” table

- dept_name is PK in the “departments” table
- role_name is PK in the “role” table
- common_email_address is assigned to a role in a specific department

➢ **benefits**

- capability to identify what department an email address belongs to and the person(s) who use it by searching the database without having to connect two tables

2) staff_salary_record: {staff_id, staff_date, type}

➢ **components**

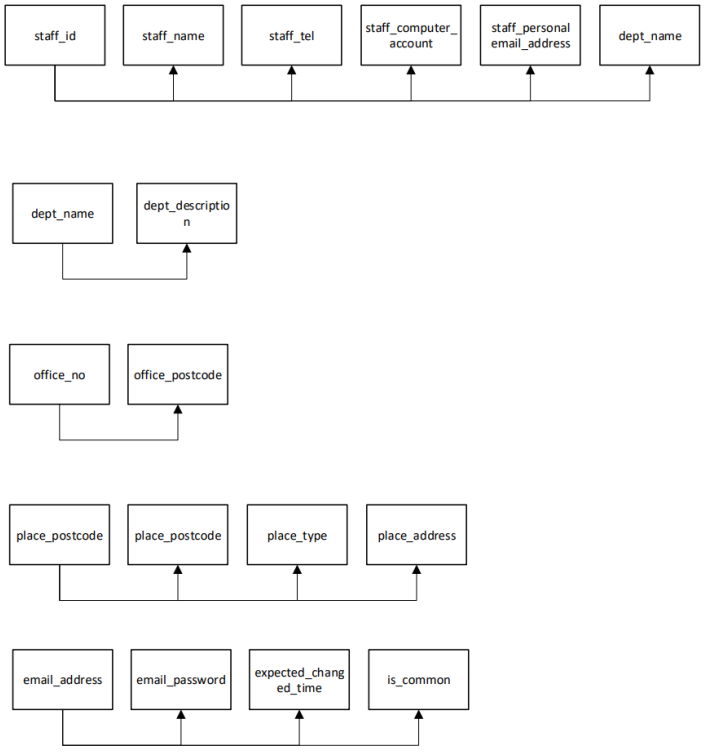
- staff_id and staff_date is the PK in the “salary” table
- type is PK in the “bonus_deduction” table

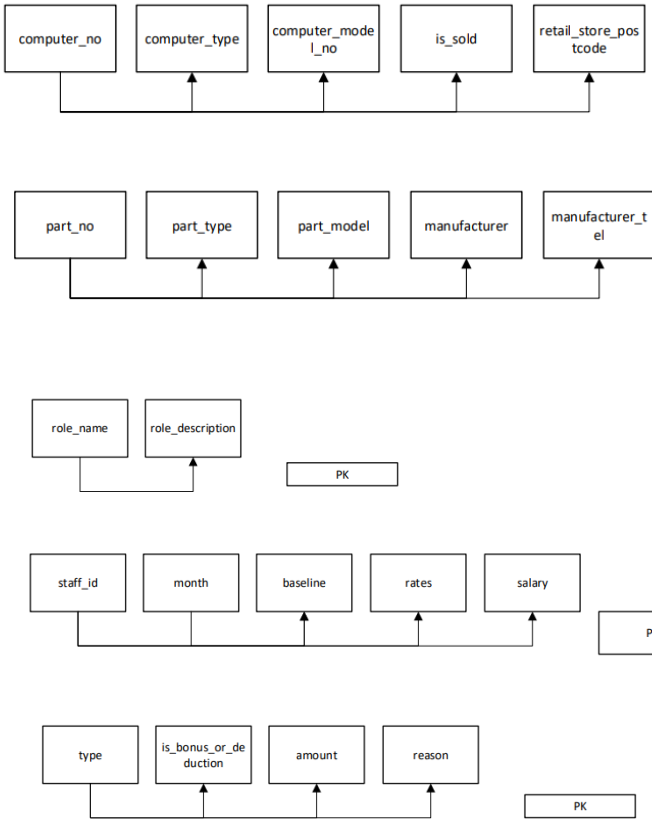
3) computer_staff: {computer_no, quality_assurrance_staff}

➢ **components**

- computer_no is PK in the “computers” table
- quality_assurance_staff is PK in the “staff” table

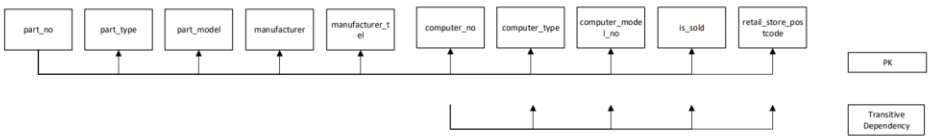
2. Functional dependency





3. Foreign key:

1) For **transitive dependency**, one table will be referenced by another table. For example,



"computers" table will be the referenced by "parts" table.

2) For **M:M relationship**, the mediate table will reference the original tables For example, "staff_dept_role" table references the "roles", "departments", "staff" table.

3) **Special case**
Sometimes, to avoid anomalies, such as inserting a wrong email, some relevant tables will adopt foreign keys.

In my design, they are as follows.

a) “dept_role_staff” table and “staff” table reference “email” table, as they both have email_address attribute, which is acquired from “email” table.
b) “computers” table references “workplaces” table, as the retail store is a workplace, as well as the HR and factory. Also, retail_store_postcode is acquired from “workplaces” table.

4. Primary key/unique key

1) Determinants in full functional dependencies can become candidate keys. Thus, for all 3NF, the determinant could be the primary key, like dept_name in the “department” table.
2) **Special case**
   I use two attributes, staff_id and staff_date as the primary key in the “salary” table. This is because staff will be paid every month and they might gain different salaries in different months. Thus, these two attributes determine the salary for this month.
3) I use some **unique key**, mainly because of the uniqueness of these attributes, like staff_tel, staff_personal_email_address.

5. Not Null

Most of the attributes should be not null, while some attributes can be null.
They are as follows:

1) It has not been decided which computer the parts will be fitted into, so computer_no in the part could be null.
2) common_email_address in the dept_role_staff can be null. This is because that “sometimes, an email address is given to a role shared by many staff members”, which means some roles may not be assigned common emails.

6. The fulfillment of particular requirements

1) Identify what department an email address belongs to and the person(s) who use it by searching the database.

   Suppose we are searching HR@durian.pc.

   ```
   SELECT DISTINCT dept_name, staff_id FROM staff_dept_role
   WHERE common_email_address = 'HR@durian.pc';
   ```

2) Find the region information by analysing the postcode

   Given that All postcodes in the south region start with “LS”, we search the staff who work in the south region.

```
SELECT staff_id FROM staff, offices
WHERE offices.office_no=staff.office_no
AND offices.office_postcode LIKE 'LS%';
```

3) Calculate the salary of every staff in a specific month

   For example, to find the bonus rates and deduction rates of staff whose ID is 123456 in December.

   - **Calculate the bonus rates**

```
SELECT SUM(bonus_deduction.amount) AS bonus_rates FROM
bonus_deduction, staff_salary_record
WHERE staff_salary_record.type=bonus_deduction.type
AND bonus_deduction.is_bonus_or_deduction='bonus'
AND staff_id = '123456'
AND month=12;
```

   - **Calculate the deduction rates**

```
SELECT SUM(bonus_deduction.amount) AS deduction_rates FROM
bonus_deduction, staff_salary_record
WHERE staff_salary_record.type=bonus_deduction.type
AND bonus_deduction.is_bonus_or_deduction='deduction'
AND staff_id = '123456'
AND month=12;
```

4) Attributes can be the IT department will ask a user to change his email password 1 year after the last time he changes his password.

   Have expected_changed_date in the “email_list” table.

   Suppose that the email is 123456@durian.pc and it will be changed Jan. 1, 2020. User entered a new password “askfghkhg123”.

```
UPDATE email_list
SET email_password = 'askfghkhg123'
WHERE email_address='123456@durian.pc'
AND expected_changed_time = '2020-01-01';
```

5) Search the information of defected parts, including the manufacturer and its tel.

   Suppose that the ID of defected part is 123456.

```
SELECT * FROM parts WHERE part_no = '123456'
```