An Online Corporate Operational Management Platform Based on Web Database*

Xingjie He[a], Mengqi Su[a], Mengjie Chen[b], Jinrui Lin[b], Yan Tong[b], Yangzhixin Luo[b], Haoxuan Xie[b], Wenke Zhong[b], Jie Pan[b] and Shuting Jiang[b]

[a]*School of Science and Engineering, Chinese University of Hong Kong, Shenzhen*
[b]*School of Data Science, Chinese University of Hong Kong, Shenzhen*

ARTICLE INFO

Keywords:
Cloud Database
Customized Setting
Data Dashboard
Greedy Algorithm
Key Performance Indicator

ABSTRACT

In this project, we built a web-based database that synthesizes various functions of an online operational management platform. This platform allows customers to post orders and employees to manage them. We aim to offer each customer considerate service, including package selection, customized settings, and status tracking. Another highlight of our platform is its analytics capabilities, which not only visualize key performance indicators but also provide assistance to product managers in determining the optimal production line based on the “first come, first served” principle using the greedy algorithm.

1. Background and Introduction

Our project aims at developing a web platform based on a consumer database for the **Chipanda Semiconductor Manufacturing Company**. This platform allows customers to place orders, as well as enables staff to manage those orders. The connection of the platform and web database allows it to easily accommodate future growth and changes in customer needs. In addition, the system has been designed with usability in mind so that customers can quickly and easily place their orders without any confusion or frustration. Finally, the system provides detailed reporting capabilities for both staff and consumers, which will allow them greater insight into order trends over time.

The target users of our platform are chip-manufacture corporations that need a secure, efficient, and easy-to-use system to manage customer orders. Our platform provides several advantages over traditional methods, such as:

- Increased concurrency measures to guarantee the successful payment during transmission and storage, providing scalability to accommodate future growth in demand;
- Detailed reporting capability provides greater insight into order trends;
- The system is designed with availability and transaction costs in mind, so plant managers can place orders in the most convenient and cost-effective manner.

By providing these features and benefits, we believe our platform will offer the best possible experience to both staff and customers alike.

* This work is a final project of CSC3170: Database System.
ORCID(s):

2. Database Structure

To simulate the real situation of a company, we constructed a relational database and generated some data based on real-world situations. This section will elaborate on the relational schema, construction process, and data generation process of our web database.

2.1. Data Collection and Generation

The data are generated using the random package of Python, combining certain real information of sample companies

User The name, country, province, phone number, and street addresses of 1062 users are randomly generated. There are two types of users: customers and staff. Staff is the employee of any plants belonging to his/her province. The province of a user is his/her shipping address, on default.

Plants We downloaded the plant locations of Intel China Inc., and randomly generated the location, phone number, and street addresses for them.

Machine We scratched the machine information from a foreign chip production website, including machines' names, versions, and prices. Each type of machine may have several copies located in different plants, and their corresponding plants are random.

Operation The **operation type** table records the names of each operation necessary for producing different chips. For simplicity, the name of an operation is composed of one letter and one number, for example, "A1" or "D10". A type of operation can be performed multiple times, and the **Operation** table records the starting time and ending time of each operation.

Chip A chip represents an actual integrated circuit to be produced, whose information generally contains its name, version, and price. The price of a chip is mainly below 300 USD, and it is usually increasingly expensive in the more advanced version. This information is included in the table **chip type**. Since there could be many copies of a type of chip, all chips produced by the company are recorded in the table **chip**. A chip can be processed by one plant, but it usually requires the cooperation of different machines.

Other relational tables The relational tables are generated according to some assumptions and constraints, listed as follows:

- One chip can only have one chip type. A chip type can be made by a single or multiple machines. A chip type is made from single or multiple types of operations.
- A machine has only one machine type, and one machine type corresponds to only one type of operation.
- Each operation will be done all at once. And the operations will be done based on the order. For a specific chip type, the type and order of operations are fixed. An operation may be done by multiple machines working together.
- Each user can place multiple orders, and users are classified as consumers and staff. The staff can also place an order as consumer, but the consumers don't have the rights of staff.
- Each package will contain many chips with multiple chip types. Each package is assigned to a factory capable of fulfillment.
- Each plant has single or multiple machine types, and each plant has multiple machines.

2.2. Relational Schema and E-R Diagram

Relational Schema

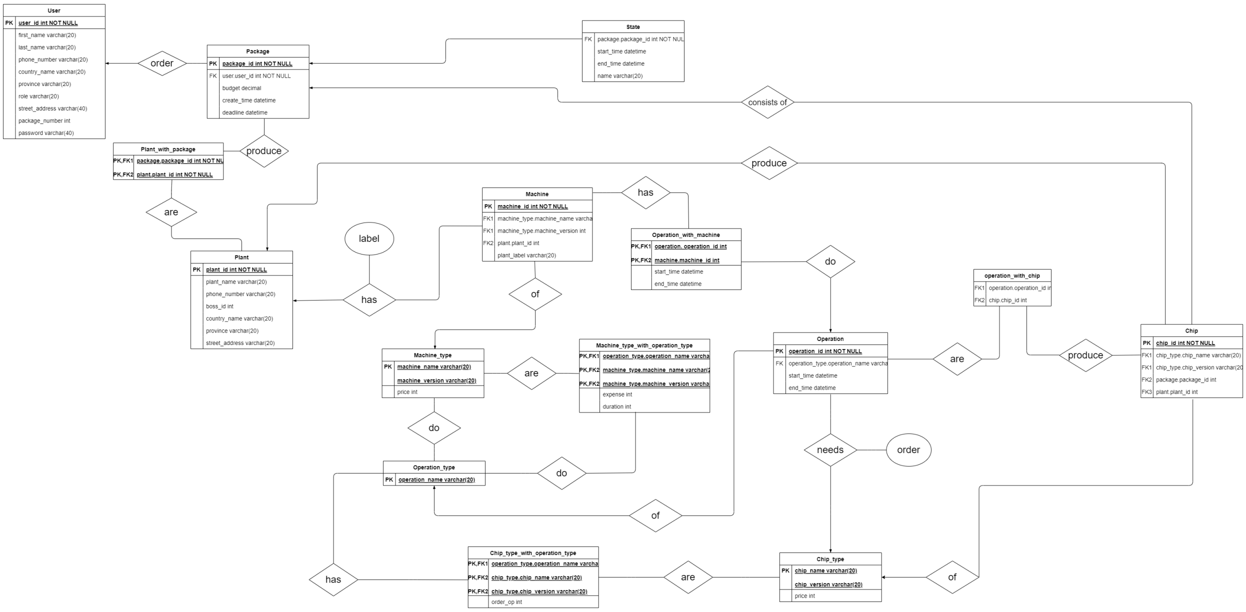

User
PK user_id int NOT NULL
first_name varchar(20)
last_name varchar(20)
phone_number varchar(20)
country_name varchar(20)
province varchar(20)
role varchar(20)
street_address varchar(40)
package_number int
password varchar(40)
order
Package
PK package_id int NOT NULL
FK user.user_id int NOT NULL
budget decimal
create_time datetime
deadline datetime
State
FK package.package_id int NOT NUL
start_time datetime
end_time datetime
name varchar(20)
consists of
produce
Plant_with_package
PK,FK1 package.package_id int NOT NU
PK,FK2 plant.plant_id int NOT NULL
produce
are
Machine
PK machine_id int NOT NULL
FK1 machine_type.machine_name varcha
FK1 machine_type.machine_version int
FK2 plant.plant_id int
plant_label varchar(20)
has
label
Operation_with_machine
PK,FK1 operation. operation_id int
PK,FK2 machine.machine_id int
start_time datetime
end_time datetime
do
Plant
PK plant_id int NOT NULL
plant_name varchar(20)
phone_number varchar(20)
boss_id int
country_name varchar(20)
province varchar(20)
street_address varchar(20)
has
of
operation_with_chip
FK1 operation.operation_id in
FK2 chip.chip_id int
Machine_type
PK machine_name varchar(20)
machine_version varchar(20)
price int
are
Machine_type_with_operation_type
PK,FK1 operation_type.operation_name varcha
PK,FK2 machine_type.machine_name varchar(2
PK,FK2 machine_type.machine_version varcha
expense int
duration int
Operation
PK operation_id int NOT NULL
FK operation_type.operation_name varcha
start_time datetime
end_time datetime
are
produce
Chip
PK chip_id int NOT NULL
FK1 chip_type.chip_name varchar(20)
FK1 chip_type.chip_version varchar(20
FK2 package.package_id int
FK3 plant.plant_id int
do
Operation_type
PK operation_name varchar(20)
do
needs
order
of
Chip_type_with_operation_type
PK,FK1 operation_type.operation_name varcha
PK,FK2 chip_type.chip_name varchar(20)
PK,FK2 chip_type.chip_version varchar(20)
order_op int
has
are
Chip_type
PK chip_name varchar(20)
chip_version varchar(20)
price int
of

Figure 1: E-R Diagram

- User(User_Id, First_Name, Last_name, Phone_number, Country_name, Province, Role, Package_done, Street_address, Password)
- Package (User_id, Package_id, Budget, Create_time, Deadline)
- State (Package_id, Start_time, End_time, State_name)
- Machine_type (Machine_name, Machine_version, Price)
- Plant (Plant_id, Plant_name, Phone_number, Boss_id, Country_name, Province, Street_address)
- Machine(Machine_name, Machine_version, Price, Plant_id, Machine_id, Plant_label)
- Chip_type (Chip_name, Chip_version, Price)
- Chip (Chip_name, Chip_version, Price, Package_id, Chip_id, Plant_id)
- Operation_type (Operation_name)
- Operation (Operation_name, Operation_id, Start_time, End_time)
- Operation_with_machine (Operation_id, Machine_id, Start_time, End_time)

The ER diagram of our system contains 15 entities. The entities are *user, package, sate, plant_with_package, machine, chip, plant, operation_with_machine, machine_type, machine_type_with_operation_type, operation, operation_type, chip_type and chip_type_with_operation_type.*

User Anyone who wants to order chips from our company will have to sign up for an account. So the user entity consists of information about the user's account, such as user's name, phone number, region, country, province, address, and role.

Package In our system, a package consists of multiple chips, so the relation between package is chips are one to many. A package must be ordered by a customer, foreign key *"user.user_id"* gives the id of the customer. *"Package_id"* identifies the package and determines other attributes. *"budget"* describes the budget for a package. *"Create_time"* is the date when the package was created, and *"Deadline"* is the date when the package was supposed to be finished.

State In our system, a package has multiple processing states, so the relation between state and package is many to one. Attribute *"Package.package_id"* is a foreign key connected to Package that identifies the package, *"Start_time"* describes the date when the package was started to be processed, *"End_time"* describes the date when the processing of the package was finished, *"Name"* describe the processing state of packages, i.e. processing; cancelled; finished.

Plant In our system, a plant consists of multiple machines, so the relation between plant and machine is many to one. Attribute *"Plant_id"* identifies the plant, *"Country_name", "Province", "Street_address"* give the location of the plant, *"Phone_number", "Boss_id"* gives the information of the manager, *"Plant_name"* gives the name of the plant.

Plant_with_package Since a plant can produce multiple packages, and a package can be produced by many plants, we designed this relationship table to describe the relationship between plant and package. *"Plant.plant_id"* refers to the plant, and *"Package.package_id"* refers to the package.

Machine In our system, a machine belongs to only one plant. *"Machine_type.machine_name"* describes the name of the machine, *"Machine_type.machine_version"* descries the version of the machine, *"Plant.plant_id"* and *"Plant_label"* describes what plant the machine belongs to.

Operation_with_machine This is a relationship table that describes the relationship between operation and machine. *"Operation.operation_id"* and *"Machine.machine_id"* refer to the corresponding operation and machine, *"State_time"* describes when the machine starts to process the operation, and *"End_time"* describes when the machine finishes the operation.

Operation Many operations are needed to produce one chip, and each operation belongs to one operation type. In our system, one operation can be processed by more than one machine.*"Operation_id"* is designed to store the ID of each operation. *"Operation_type.operation_name"* is used to determine the operation type corresponding to this operation. *"Start_time"* and *"end_time"* are designed to store the start time and end time of each specific operation.

Operation_with_chip A combination of chips and operations. Many operations will be applied when it comes to producing one chip, and one operation can be applied to many chips, the relation is many to many. *chip.chip_id* gives the ID of the chip, and *operation.operation_id* gives the ID of the operation.

Operation_type In our system, an operation type is determined, if and only if for any machine, the time and expense to process it is determined. Typical operation types are "design-import", "etch", "drill" etc. The attribute *"operation_name"* is designed to store the name of each operation.

Machine_type Each machine belongs to one specific machine type. One machine type has corresponding specific operation types. *"Machine_type.machine_name"* and *"machine_type.machine_version"* are designed to store the name and version identifier of each machine, and the latter one cannot be empty. *"Price"* is designed to store the price of each machine.

Machine_type_with_operation_type In our system, each type of machine can only process specific operation types. This entity has five attributes. *"Operation_type.operation_name"* is designed to store the name of each operation. *"Machine_type.machine_name"* and *"machine_type.machine_version"* are designed to store the name and version of each machine. "Expense" is designed to store the expense that a certain type of machine needs to process a certain type of operation. "Duration" is designed to store the time a certain type of machine needs to process a certain type of operation.

Chip *"Chip_id"* is designed to store the ID of each chip. This entity has four foreign keys of other entities. *"Chip_type.chip_name"* and *"Chip_type.chip_version"* are designed to link the chip with a corresponding chip. *"Package.package_id"* is designed to determine which package the chip is from. *"Plant.plant_id"* is designed to determine which plant the chip is to be produced in.

Chip_type Each chip belongs to one specific chip type. *"Chip_name"* is designed to store the name of each chip, and it cannot be empty. *"Chip_version"* is designed to store the production version identifier of each chip, and it cannot e empty. *"Price"* is designed to store the sale price of each chip type, and the chip price is determined by this attribute.

Chip_type_with_operation_type If the chip type is given, the operation type should be determined. One chip type needs more than one operation. In our system, there aren't repeated operation types when processing one chip type. *"Operation_type.operation_name"* is designed to store the name of each operation. *"Chip_type.chip_name"* and *"chip_type.chip_version"* are designed to store the name and production version of each chip type. *"Order_op"* is designed to store the order of the operation in chip manufacturing.

2.3. Cloud Database

We utilize the Huawei cloud database RDS for MySQL to be the back-end database of our project. The version of the database engine is MySQL 5.7.38. The RDS instance is equipped with 2 vCPUs and 4 GB of main memory.

After the initialization of the cloud database, we create the schema and tables using *source/db_table.sql*, then insert the data by directly loading the CSV datasets, since the column names in the CSV datasets conform to those in the creating-table SQL respectively. *source/db_insert.sql* is for locally testing whether the insertion of data works as expected. For debugging purposes, we also provide *source/db_clean.sql* to drop all the database tables.

The public IP address of our RDS instance is 123.60.157.95, with the port being 3306. We have granted access right from all IP addresses, thus the front end can retrieve the database structure and data using the following code:

```
cnx = mysql.connector.connect(
    host="123.60.157.95",
    port=3306,
    user="root",
    password="xxx", # replace the password here
    database="project")
```

The schema management UI interface of Huawei cloud database can refer to the following:
The SQL execution UI interface of Huawei cloud database can refer to the following:

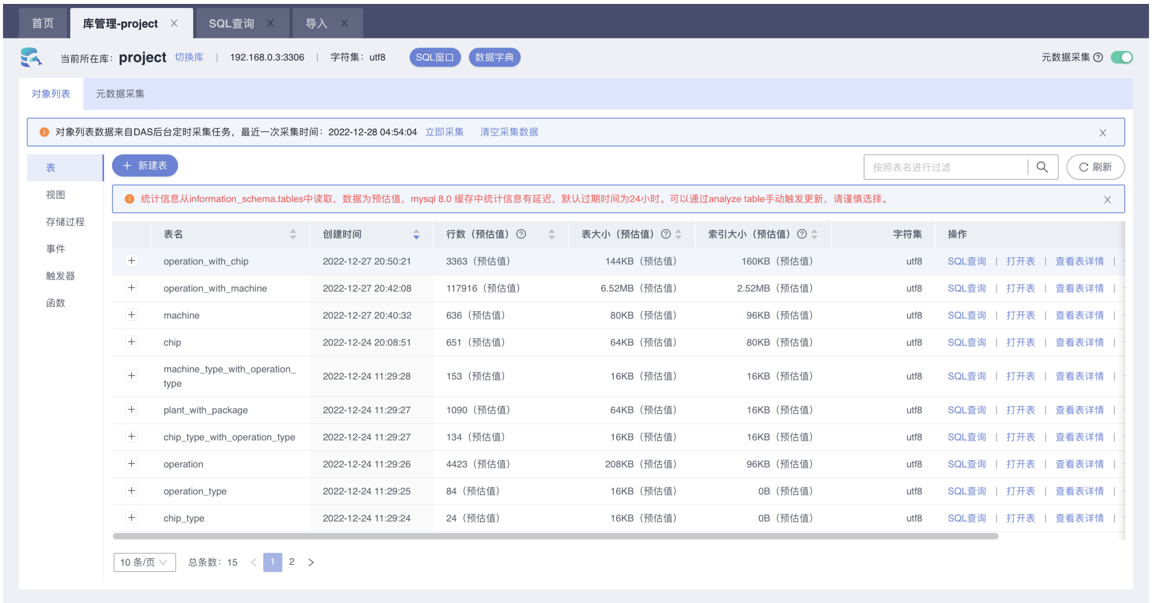
3. General Functions

Our company is named Chipanda Semiconductor Manufacturing Company, and we have developed a web platform to facilitate communication and collaboration with both staff and consumers. The platform consists of three main sections: the login page, the staff portal, and the consumer portal. Here is a brief introduction of those three parts.

- **LOGIN PAGE** Upon accessing the platform, users are directed to the login page where they can enter their registered credentials. Based on their identity, they are then redirected to the appropriate portal. New users can register for an account on this page as well.

- **CONSUMER PORTAL** The consumer portal allows consumers to order chips and offers information about their current packages.

- **STAFF PORTAL** The staff portal is designed for company employees, on which they can access information about plants and assign operations to different packages.



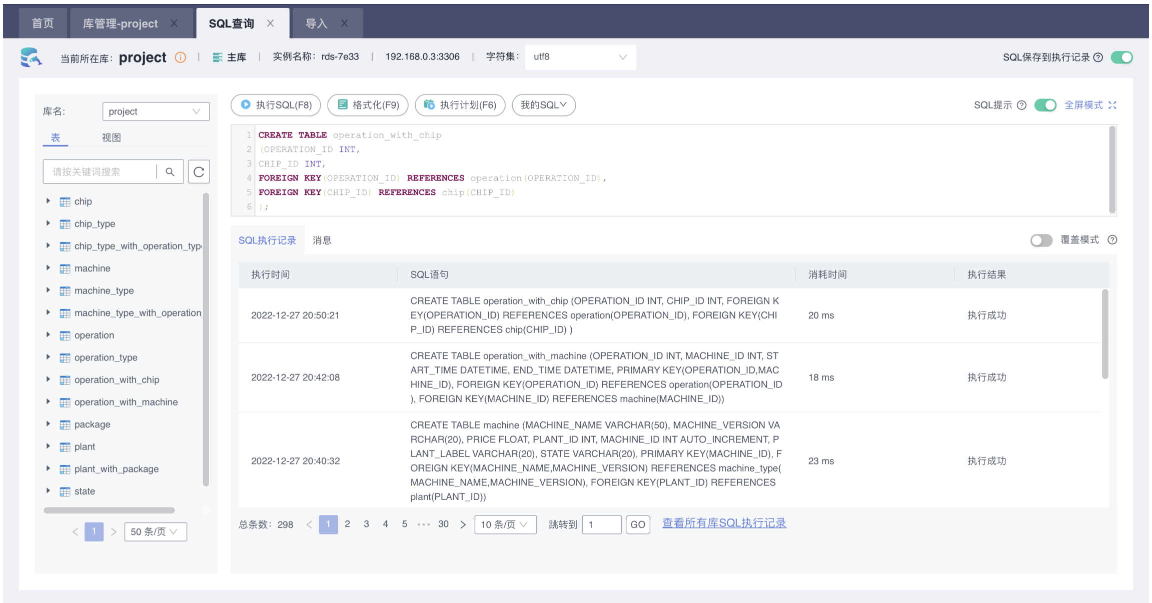
对象列表数据来自DAS后台定时采集任务，最近一次采集时间：2022-12-28 04:54:04 立即采集 清空采集数据

统计信息从information_schema.tables中读取，数据为预估值，mysql 8.0 缓存中统计信息有延迟，默认过期时间为24小时。可以通过analyze table手动触发更新，请谨慎选择。

	表名	创建时间	行数（预估值）	表大小（预估值）	索引大小（预估值）	字符集	操作
＋	operation_with_chip	2022-12-27 20:50:21	3363（预估值）	144KB（预估值）	160KB（预估值）	utf8	SQL查询 | 打开表 | 查看表详情 |
＋	operation_with_machine	2022-12-27 20:42:08	117916（预估值）	6.52MB（预估值）	2.52MB（预估值）	utf8	SQL查询 | 打开表 | 查看表详情 |
＋	machine	2022-12-27 20:40:32	636（预估值）	80KB（预估值）	96KB（预估值）	utf8	SQL查询 | 打开表 | 查看表详情 |
＋	chip	2022-12-24 20:08:51	651（预估值）	64KB（预估值）	80KB（预估值）	utf8	SQL查询 | 打开表 | 查看表详情 |
＋	machine_type_with_operation_type	2022-12-24 11:29:28	153（预估值）	16KB（预估值）	16KB（预估值）	utf8	SQL查询 | 打开表 | 查看表详情 |
＋	plant_with_package	2022-12-24 11:29:27	1090（预估值）	64KB（预估值）	16KB（预估值）	utf8	SQL查询 | 打开表 | 查看表详情 |
＋	chip_type_with_operation_type	2022-12-24 11:29:27	134（预估值）	16KB（预估值）	16KB（预估值）	utf8	SQL查询 | 打开表 | 查看表详情 |
＋	operation	2022-12-24 11:29:26	4423（预估值）	208KB（预估值）	96KB（预估值）	utf8	SQL查询 | 打开表 | 查看表详情 |
＋	operation_type	2022-12-24 11:29:25	84（预估值）	16KB（预估值）	0B（预估值）	utf8	SQL查询 | 打开表 | 查看表详情 |
＋	chip_type	2022-12-24 11:29:24	24（预估值）	16KB（预估值）	0B（预估值）	utf8	SQL查询 | 打开表 | 查看表详情 |

10 条/页 总条数：15 < 1 2 >

Figure 2: The schema management UI interface of the Huawei cloud database



- chip
- chip_type
- chip_type_with_operation_typ
- machine
- machine_type
- machine_type_with_operation
- operation
- operation_type
- operation_with_chip
- operation_with_machine
- package
- plant
- plant_with_package
- state

```
CREATE TABLE operation_with_chip
(OPERATION_ID INT,
CHIP_ID INT,
FOREIGN KEY(OPERATION_ID) REFERENCES operation(OPERATION_ID),
FOREIGN KEY(CHIP_ID) REFERENCES chip(CHIP_ID)
);
```

执行时间	SQL语句	消耗时间	执行结果
2022-12-27 20:50:21	CREATE TABLE operation_with_chip (OPERATION_ID INT, CHIP_ID INT, FOREIGN KEY(OPERATION_ID) REFERENCES operation(OPERATION_ID), FOREIGN KEY(CHIP_ID) REFERENCES chip(CHIP_ID))	20 ms	执行成功
2022-12-27 20:42:08	CREATE TABLE operation_with_machine (OPERATION_ID INT, MACHINE_ID INT, START_TIME DATETIME, END_TIME DATETIME, PRIMARY KEY(OPERATION_ID,MACHINE_ID), FOREIGN KEY(OPERATION_ID) REFERENCES operation(OPERATION_ID), FOREIGN KEY(MACHINE_ID) REFERENCES machine(MACHINE_ID))	18 ms	执行成功
2022-12-27 20:40:32	CREATE TABLE machine (MACHINE_NAME VARCHAR(50), MACHINE_VERSION VARCHAR(20), PRICE FLOAT, PLANT_ID INT, MACHINE_ID INT AUTO_INCREMENT, PLANT_LABEL VARCHAR(20), STATE VARCHAR(20), PRIMARY KEY(MACHINE_ID), FOREIGN KEY(MACHINE_NAME,MACHINE_VERSION) REFERENCES machine_type(MACHINE_NAME,MACHINE_VERSION), FOREIGN KEY(PLANT_ID) REFERENCES plant(PLANT_ID))	23 ms	执行成功

总条数：298 < 1 2 3 4 5 ••• 30 > 10 条/页 跳转到 1 GO 查看所有库SQL执行记录

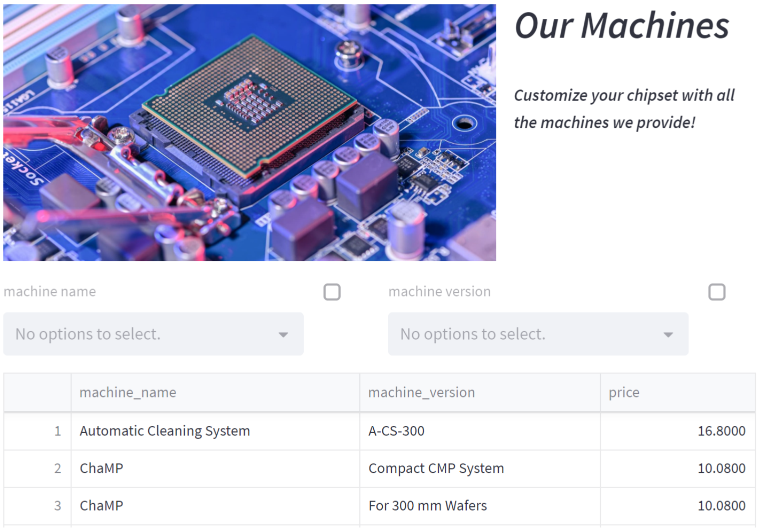
Figure 3: The SQL execution UI interface of the Huawei cloud database

3.1. Log-in Page

The log-in interface is the gateway to our web platform. It allows users to access either the STAFF portal or the CONSUMER portal based on their registered identity. To log in, users simply need to enter their first name, last name, and password. Once logged in, they will be directed to the appropriate portal where they can access the resources and tools they need. New users can register for an account on this page as well. The log-in interface is designed to be user-friendly and easy to use, ensuring that users can quickly and easily access the information they need. It is an essential part of our platform, helping to streamline communication and workflow within the staff, as well as providing a convenient interface for our clients to access information and resources.

3.2. Main Page

The main page of both customer and staff accounts will present the overall picture of the Chipanda company. The information displayed on this page includes the plant distribution map, the product list, and the device list.



Our Machines

Customize your chipset with all the machines we provide!

	machine_name	machine_version	price
1	Automatic Cleaning System	A-CS-300	16.8000
2	ChaMP	Compact CMP System	10.0800
3	ChaMP	For 300 mm Wafers	10.0800

Figure 4: A part of the main page

3.3. 2C: Consumer Interface

Through the CONSUMER portal, users can access a wide range of features and functions, including the ability to purchase any of the over 20 different types of chips offered by our company. The CONSUMER portal is constantly updated with the latest chips as they are released, making it easy for users to keep track of the latest products. In addition to purchasing chips, users can also specify which factory they would like to use for processing, with shipping costs calculated based on the distance between the factory and the user. The CONSUMER portal also provides a convenient way for users to track and supplement their orders.

3.3.1. Shopping cart page

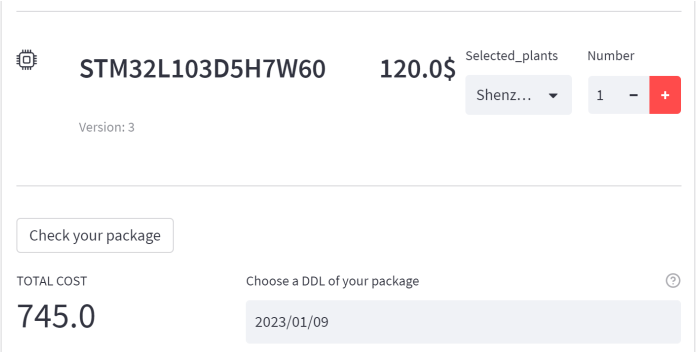
The shopping cart page of the platform allows users to submit their package information and make a purchase. On this page, users can review the items they have added to their cart and make any necessary adjustments before completing their order. The proper execution step is as follows.

1. Select the products and specify customized settings: Pick a plant as the manufacturer and set the number of the chip.

2. Click the bottom "check the package". The total price will be displayed. Then, you are prompted to set the deadline, which should be 10 days at least and 30 days at most from now.

3. May payments and submit the order.

There are a total of 20+ different types of chips on sale. For each product, the user can choose one available plant to produce the chip. After the user submits the package, the information in the cloud database



STM32L103D5H7W60 120.0$

Version: 3

Selected_plants: Shenz...

Number: 1

TOTAL COST

745.0

Choose a DDL of your package

2023/01/09

Figure 5: The order-submission interface

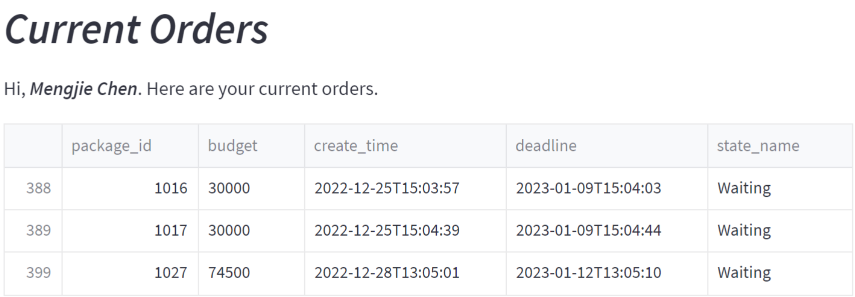
about chip and package will be updated. Users can check those updated information on the current order page as well. The shopping cart page is an important step in the chip shopping process, as it allows users to review and confirm their purchases before finalizing their orders. Overall, the shopping cart page is designed to make the process of buying chips on the platform as easy and convenient as possible.

3.3.2. Current order page

The current order page has 3 functions in total. First, the current order page shows the current orders of the user. On this page, user can easily view their ordered packages with their respective status: Waiting, In Process, Finished, or Cancelled. Once the status is changed by the staff, the corresponding package status on the customer-side webpage will be synchronized.

Canceling a package is equivalent to canceling all the operations of all the chips in that package, rather than deleting some particular chips in that package. Users are also allowed to cancel the order as long as this order has not been processed. Once a chip is processed, the package can no longer be cancelled.

The information of all plants working for the user will be shown along with the current order, including the street address and phone number. This function aims to provide direct after-sales service since the customer can easily contact the staff of the plant. The locations of these plants are also plotted on the map, shown as figure 7.



Current Orders

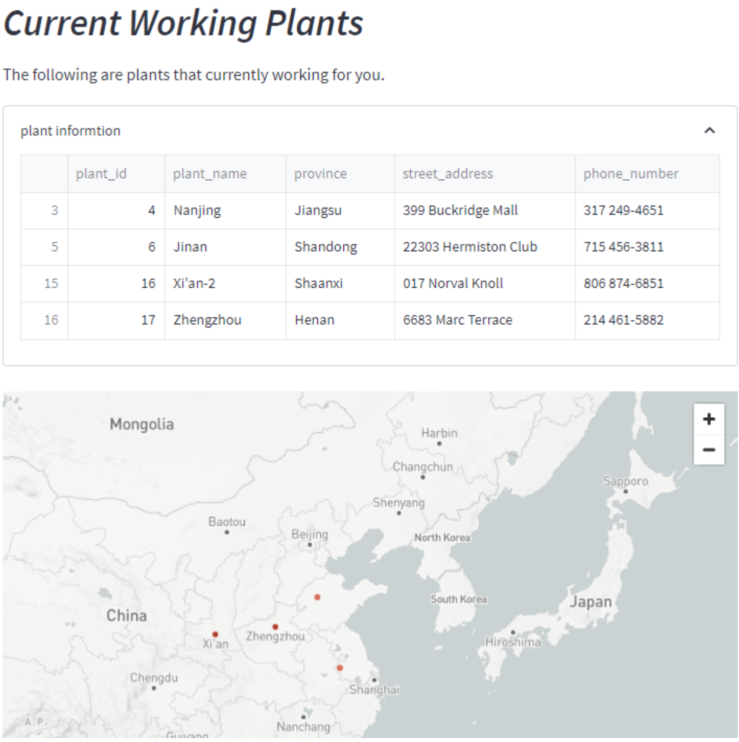
Hi, ***Mengjie Chen***. Here are your current orders.

	package_id	budget	create_time	deadline	state_name
388	1016	30000	2022-12-25T15:03:57	2023-01-09T15:04:03	Waiting
389	1017	30000	2022-12-25T15:04:39	2023-01-09T15:04:44	Waiting
399	1027	74500	2022-12-28T13:05:01	2023-01-12T13:05:10	Waiting

Figure 6: Show current orders and cancel waiting orders

3.4. 2B: Staff management System

The STAFF portal on Chipanda Semiconductor Manufacturing Company Limited's web platform is a valuable resource for our company's employees. Through the STAFF portal, employees can access a range of tools and features to manage and streamline their work. For example, STAFF can use the portal



Current Working Plants

The following are plants that currently working for you.

plant informtion

	plant_id	plant_name	province	street_address	phone_number
3	4	Nanjing	Jiangsu	399 Buckridge Mall	317 249-4651
5	6	Jinan	Shandong	22303 Hermiston Club	715 456-3811
15	16	Xi'an-2	Shaanxi	017 Norval Knoll	806 874-6851
16	17	Zhengzhou	Henan	6683 Marc Terrace	214 461-5882

Figure 7: Show plants that are working for you

to assign machines and use the operation function to process packages. The STAFF portal also provides a convenient way for employees to view performance data for individual factories and track the progress of their work. Overall, the STAFF portal is a powerful tool that helps our company's employees stay organized and productive.

3.4.1. Plant management page

This page contains two main functions. First, the staff can manage the chips. Staff can select any number of chips that are in the state of **"Waiting"**, and the system will assign them machines according to the **Greedy Algorithm**, which will be elaborated in section 4.2. After clicking the submit button, the database will be updated during this process, so there should be a page refresh to observe the results after management. Eventually, these selected chips will be in the state of **"Processing"** and therefore will not appear in the selection box of the first function. The second function, on the other hand, displays the uncompleted chips in that plant in a table. Whenever an employee completes managing a chip under the first function, the chip will enter the state of "Processing" and will be displayed in the table.

3.4.2. Plant performance page

This page is used to show the company's performance for a year. It has a total of three data displays. The first display is the **top six chips by REVENUE** in each province. Users can select any number of provinces in a multi-select box, and the six highest-selling chips in the selected provinces will be displayed in a table.

The second display is the change in **the company's REVENUE** for each month, displayed in a line chart. Staff can use this table to view the company's monthly revenue status.

The third display is the change in the number of **orders accepted** and the number of **orders completed** by the company for each month, displayed in an area chart. Staff can use this chart to check the number of orders received by the company for each month and the status of their completion.

4. Analytical Functions

Our platform provides various analytical functions for chip production companies to maximize their efficiency. By coordinating the SQL query and the plotting functions, some key indexes can be visualized as graphs. A more advanced application of analytics is the automatic machine allocation function, where the greedy algorithm helps allocates the production task to available machines while updating the database information.

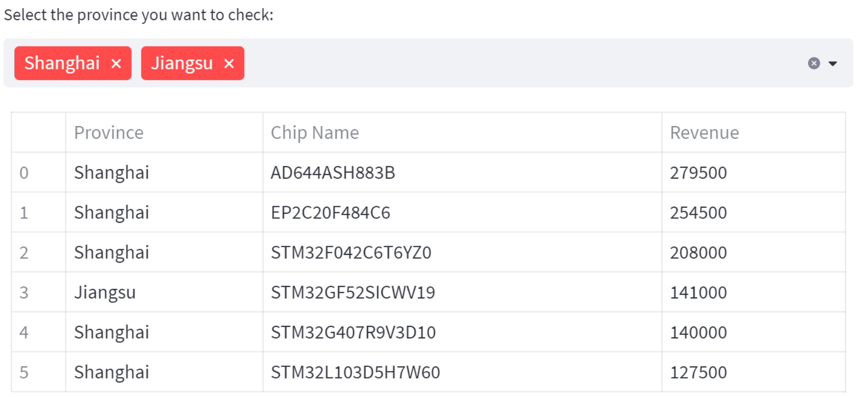
4.1. Data Visualization Dashboard

Data visualization capabilities offer a comprehensive overview of sales, customer relations, and delivery status throughout the country, helping with identifying areas of improvement. On the staff side, we offer three categories to help the plant managers understand the sales performance. With our intuitive interface and powerful analytical tools, businesses can quickly make informed decisions about their operations in order to increase productivity and reduce overhead expenses.

4.1.1. Marketable products in different regions

To see the list of chips that brings about the maximal revenue, the user can filter the provinces in the select box. Then, the most profitable products (up to 6 items) will be listed in a table, shown as figure 10. This function allows multi-selection, similar to the situation where a regional sales manager wants to inquire the cross-province sales within his region.

```
SELECT u.province, c.chip_name AS ChipName, SUM(p.budget) AS revenue
FROM user AS u NATURAL JOIN package AS p, chip AS c
WHERE p.package_id=c.package_id AND province IN (%s)
GROUP BY c.chip_name
ORDER BY SUM(p.budget) DESC;
```



Select the province you want to check:

Shanghai × Jiangsu ×

	Province	Chip Name	Revenue
0	Shanghai	AD644ASH883B	279500
1	Shanghai	EP2C20F484C6	254500
2	Shanghai	STM32F042C6T6YZ0	208000
3	Jiangsu	STM32GF52SICWV19	141000
4	Shanghai	STM32G407R9V3D10	140000
5	Shanghai	STM32L103D5H7W60	127500

Figure 8: Most profitable products in Shanghai plus Jiangsu

4.1.2. Revenue time series

A line chart indicates the fluctuation of revenue with respect to the time horizon. From the chart, the staff can clearly perceive the trend of sales performances in the past months. The chart update is synchronized with the database records.

```
SELECT SUM(p.budget) AS revenue, DATE_FORMAT(p.create_time, '%%Y-%%m-%%d') AS
    create_time
FROM user AS u, package AS p
WHERE u.province IN (%s)
GROUP BY MONTH(p.create_time)
ORDER BY p.create_time ASC;
```

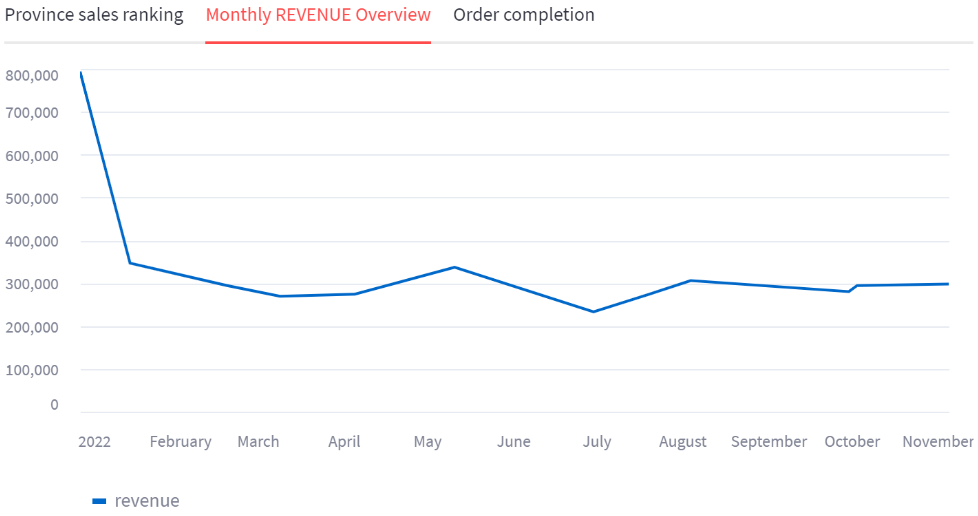


Figure 9: Time series graph of total revenue

4.1.3. Order completion rate

An important index to evaluate a staff's performance is his/her completion rate of orders. Usually, we want this index to be as high as possible, which means that the staff possesses a strong ability to deal with numerous orders. Due to the time limitation, we do not display this index for individual staff separately, but only discuss the overall completion rate.

```
-- The number of completed orders per month:
SELECT create_date, COUNT(*) AS finish_num
FROM (SELECT DATE(p.create_time) AS create_date, s.state_name AS status
    FROM package AS p NATURAL JOIN state AS s) AS joint
WHERE status='Finished'
GROUP BY MONTH(create_date)
ORDER BY create_date ASC;

-- The number of TOTAL orders, including finished and cancelled ones:
SELECT create_date, COUNT(*) AS all_num
FROM (SELECT DATE(p.create_time) AS create_date, s.state_name AS status
    FROM package AS p NATURAL JOIN state AS s) AS joint
GROUP BY MONTH(create_date)
ORDER BY create_date ASC;
```

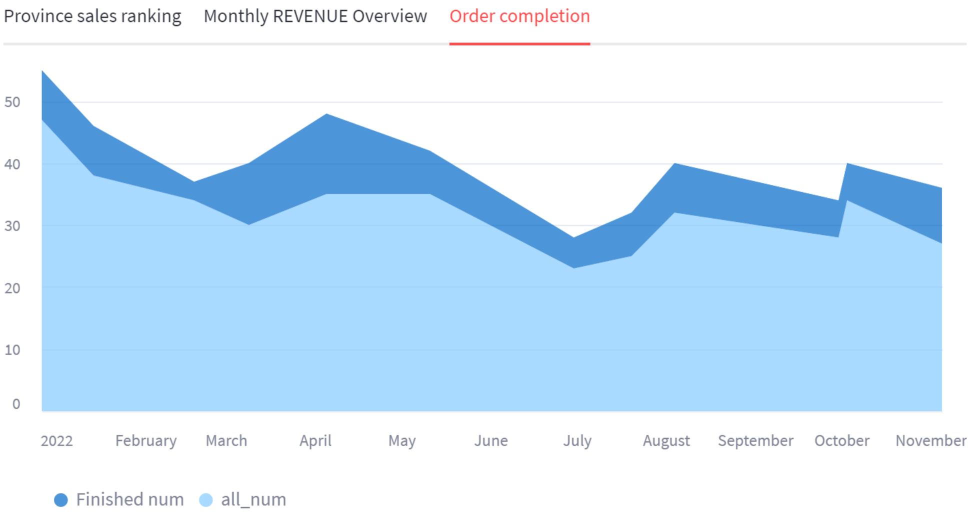
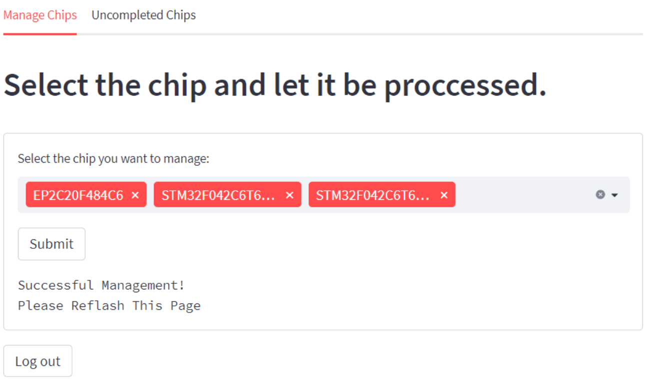


Figure 10: Order completion rate, shown in an area chart

4.2. Greedy Algorithm for Machine Allocation

The consumer could assign each chip in the package to a plant. After that, the chip would be allocated to the machines of the plant automatically. Once a staff in the corresponding plant sees the unprocessed order and is ready to deal with it, he/she only needs to select them and click "submit" to submit the processing request. Then, our algorithm will automatically allocate the manufacture task to suitable machines.



Select the chip and let it be proccessed.

Select the chip you want to manage:

EP2C20F484C6 × STM32F042C6T6... × STM32F042C6T6... ×

```
Successful Management!
Please Reflash This Page
```

Figure 11: The staff can submit the processing request here

Assuming each operation corresponds to one machine assignment, we developed a greedy algorithm for the machine allocation. The greedy algorithm generally follows “first come, first served” principle. That is, for each operation, the algorithm finds the first available (i.e. FREE) machine and assigns the operation to that machine. Then, the state of the machine is set as "busy". The detailed procedure is shown in Algorithm 1.

Algorithm 1: Machine-Allocation(chip_id)

```
Find the next operation type $t$ to process the chip;
Find the plant $p$ assigned to the chip;
for each machine $m$ in the plant $p$ do
    if machine $m$ is FREE then
        Find all operation types $T$ the machine could process;
        if $t \in T$ then
            Generate a new operation for the chip;
            Generate a relation between the chip and the operation;
            Generate a relation between the machine and the relation;
            Set machine $m$ as BUSY;
            Break the loop;
```

5. Conclusion

The project intends to build a web-based database that synthesizes various functions of an online operational management platform. With this goal, on the back end, the project members finished building the database in the cloud and created the underlying schemas. In addition, the data was generated. On the front end, the web page allows customers to log in with an account or register an account. After logging in, the webpage provides customers with an introduction page, an order placement interface, and an order management interface. The site also provides staff with pages to check the company's performance and manage packages. In this project, the team members have accomplished achievements in database generation, database access, and database update.

6. Contribution

Every team member contributed to the project equally.