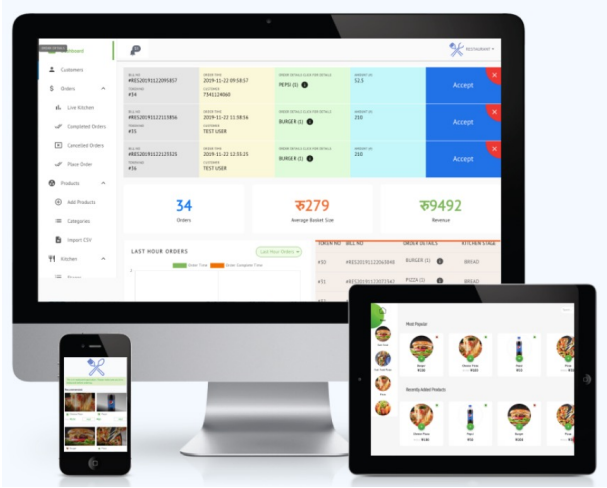
Restaurant Order Management System

Agenda

- Introduction
- What is Entity Relation
- ER Diagram Explanation
- SQL Queries and Algebra Relationship and Output
- Conclusion

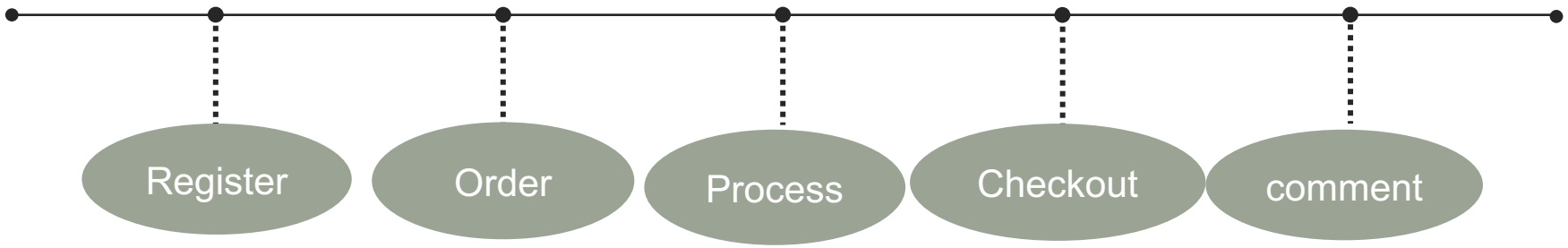
Introduce



The Business can:

- Manage menu: add, delete, modify, query.
- Manage employee information.
- Process order: serve for customer and process checkout.
- Recording payment: monthly bill and year bill.

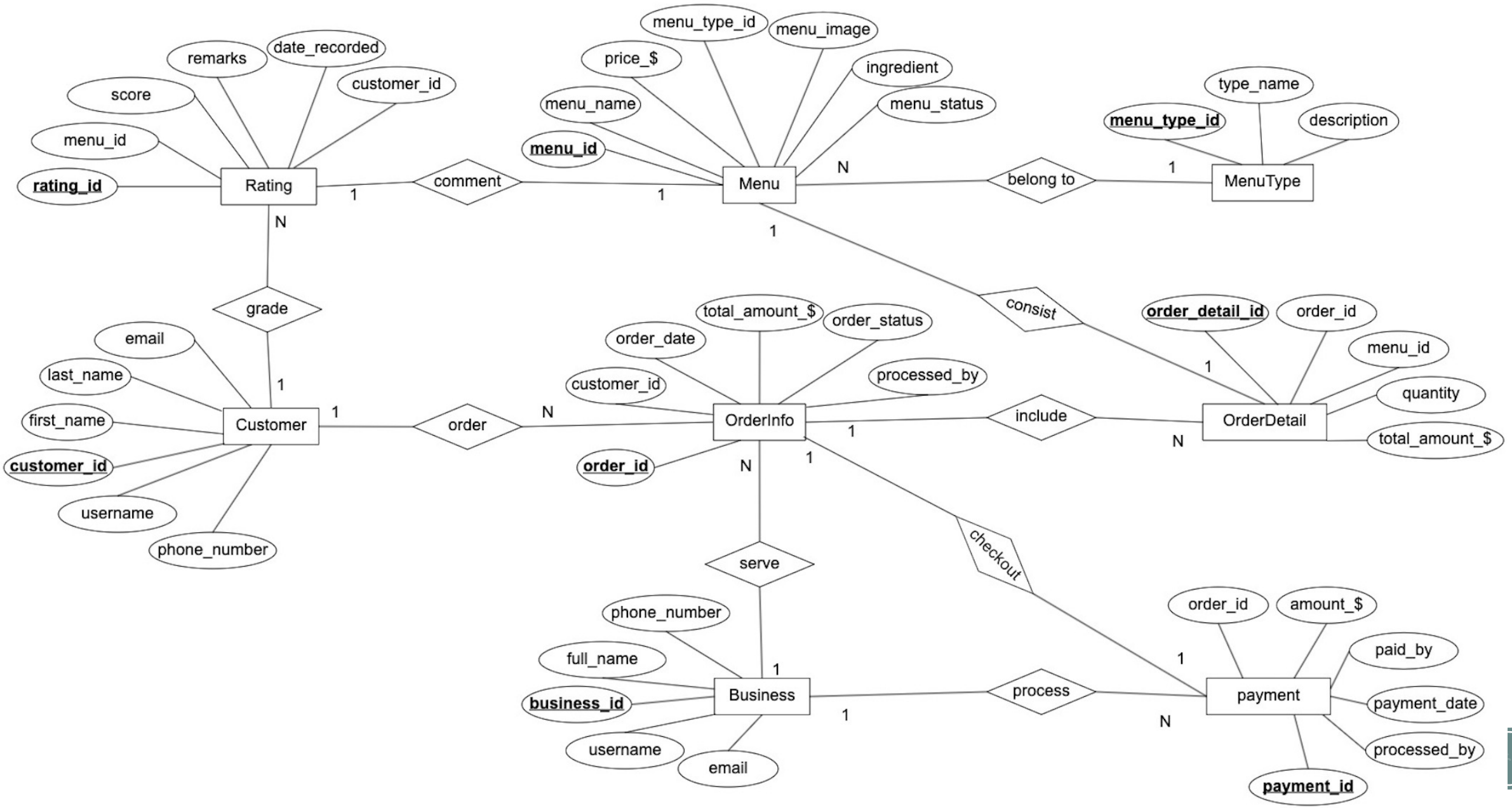
- **The Customer can:**
  - Order the menu.
  - Track order: order status, order detail and history order information.



What Is Entity Relation Modeling

- Entity Relationship Model is a **graphical approach to database design**;
- It is a high-level data model that defines data elements and their relationship for a specified software system;
- An ER model is used to represent **real-world objects**.
  - An entity has a set of properties.
  - Entity properties can have values.

ER Diagram


remarks
date_recorded
score
customer_id
menu_id
rating_id
Rating
comment
menu_type_id
menu_image
price_$
ingredient
menu_name
menu_status
menu_id
Menu
belong to
type_name
menu_type_id
description
MenuType
grade
consist
email
last_name
first_name
customer_id
username
phone_number
Customer
order
total_amount_$
order_status
order_date
customer_id
processed_by
OrderInfo
order_id
include
order_detail_id
order_id
menu_id
quantity
total_amount_$
OrderDetail
serve
checkout
phone_number
full_name
business_id
username
email
Business
process
order_id
amount_$
paid_by
payment_date
processed_by
payment_id
payment
1
N

ER diagram Explanation

Entity Types

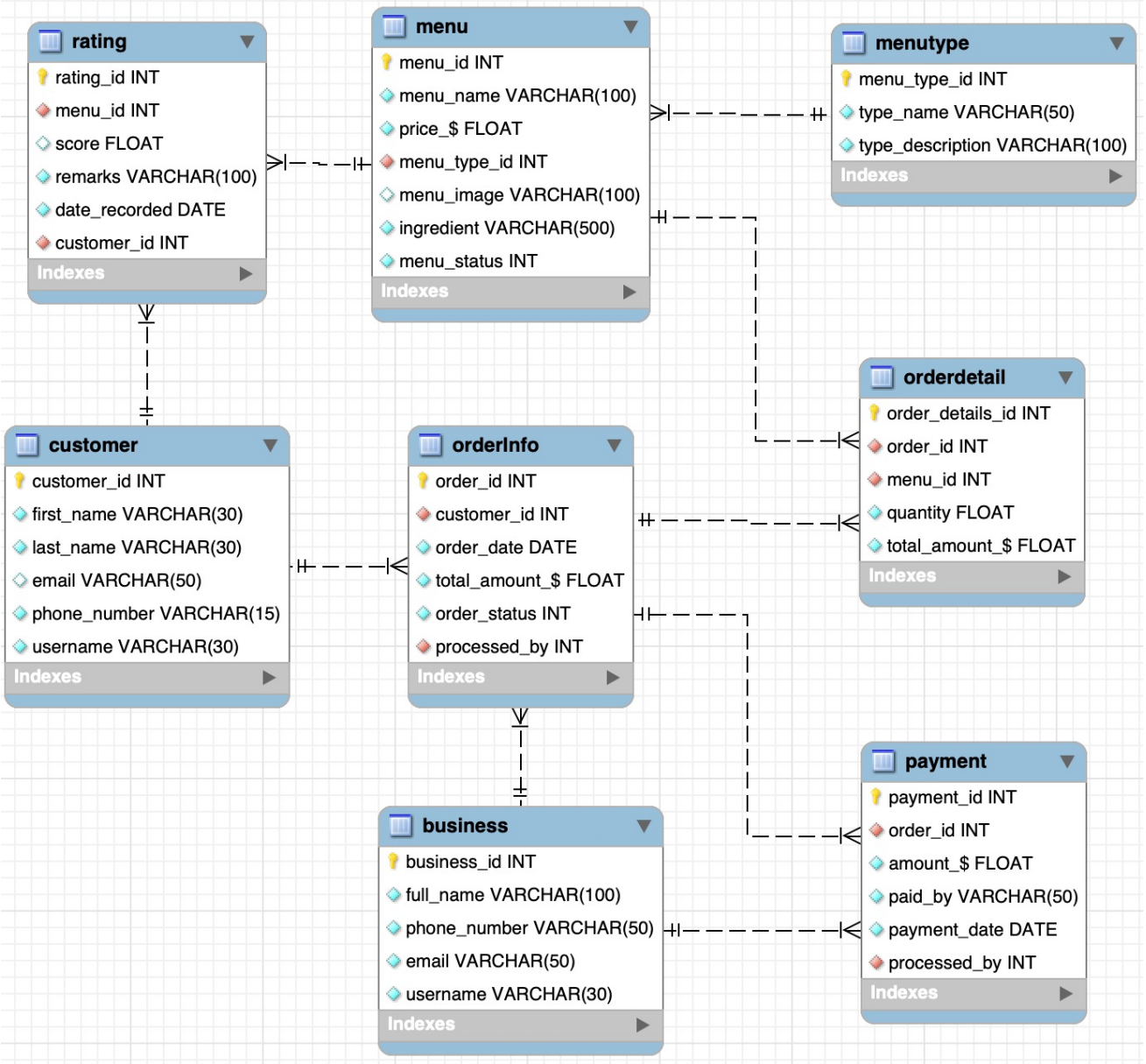
- **Customer**: customer_id, first_name, last_name, email, phone_number, profile_image, username, psword
- **Business**: business_id, full_name, phone_number, email, username, psword
- **Menu**: menu_id, menu_name, price, menu_type, menu_image, ingredients, menu_status
- **Menutype**: menu_type_id, type_name, type_description
- **OrderInfo**: order_id, customer_id, order_date, total_amount, order_status, processed_by
- **Orderdetail**: order_details_id, order_id, menu_id, amount, no_of_serving, total_amount
- **Payment**: payment_id, order_id, amount, paid_by, payment_date, processed_by
- **Rating**: rating_id, menu_id, score, remarks, date_recorded, customer_id

ER diagram Explanation

Relationship Types

- **Order**: Customer order food, which has cardinality of 1:N, which means the customer can order many times.
- **Include**: The OrderInfo include many orderDetail. Cardinality ratio is 1:N
- **Consist**: The orderDetail is consist by menu, Cardinality ratio is 1:1
- **Belong to**: A menu belong to a specific MenuType, Cardinality ratio is N:1
- **Serve**: Business will serve for use, so the Cardinality ratio is 1:N
- **Checkout**: When user finishes their meal, they need checkout, the Cardinality between orderInfo and payment is 1:1
- **Process**: When user checkout, Business need process the payment. The Cardinality ratio is 1:N
- **Grade**: After user checkout, they could grade to the menu, the Cardinality between customer and rating is 1:N
- **Comment**: This relationship relates Rating and Menu. The Cardinality ratio is 1:1 as multiple customer can comment same menu.

Schema structure



Schema

customer

customer_id	first_name	last_name	email	phone_number	username
1	Nacy	Smith	smith@gmai.com	1 888-348-9045	chen
2	William	Brown	brown@gmai.com	1 888-601-5957	brown
3	Emily	Jones	jones@gmai.com	1 877-457-9044	jones
4	Jimmy	Henry	henry@gmai.com	1 866-125-8043	henry
5	James	Johnson	johnson@gmai.com	1 855-380-4047	johnson
6	Json	Chang	chang@gmai.com	1 844-894-3042	change
7	Emily	jing	jing@gmai.com	1 833-234-8049	jing
8	John	ding	ding@gmai.com	1 822-601-6043	ding

business

business_id	full_name	phone_number	email	username
1	Lily Lee	1 674-823-4089	lily@gmail.com	lily
2	Jeff Davis	1 625-828-9045	jeff@gmail.com	jeff
3	Linda Smith	1 925-563-6525	linda@gmail.com	lida
4	Monica Wilson	1 925-453-9999	wilson@gmail.com	wilson

Schema

Menu

menu_id	menu_name	price_$	menu_type_id	menu_image	ingredient	menu_status
1	Twice Cooked Pork	19.9	1	http://localhost:8080/1	pork	1
2	Spicy Fish	39.9	2	http://localhost:8080/2	fish	0
3	Tomato Egg Soup	9.9	3	http://localhost:8080/3	tomato	0
4	Dandan Noodles	15.9	4	http://localhost:8080/4	flour	1
5	Egg Roll	19.9	1	http://localhost:8080/5	egg	1

MenuType

menu_type_id	type_name	type_description
1	Hubei Cuisine	It is well known for its hot and spicy flavours.
2	Hunan Cuisine	Hubei cuisine is derived from styles of Hubei Province in China.
3	Beijing Cuisine	Beijing cuisine, also known as Jing cuisine.
4	Shanghai Cuisine	Shanghai cuisine emphasises the use of condiments.

Schema

OrderInfo

order_id	customer_id	order_date	total_amount_$	order_status	processed_by
1	1	2020-12-12	19.9	0	1
2	2	2020-11-19	59.7	0	2
3	3	2020-10-14	19.9	1	3
4	4	2020-09-16	39.9	1	4
5	5	2020-05-12	55.7	1	3
6	6	2020-03-28	39.9	1	2

OrderDetail

order_details_id	order_id	menu_id	quantity	total_amount_$
1	1	1	1	19.9
2	2	2	1	39.9
3	2	3	2	19.8
4	3	5	1	19.9
5	4	2	1	39.9
6	5	4	1	15.9
7	5	1	1	19.9
8	6	2	1	39.9

Schema

Payment

payment_id	order_id	amount_$	paid_by	payment_date	processed_by
1	3	19.9	Peter	2020-10-14	2
2	4	39.9	Yang	2020-09-16	1
3	5	55.7	Chang	2020-05-12	2
4	6	39.9	Ken	2020-03-28	2

Rating

rating_id	menu_id	score	remarks	date_recorded	customer_id
1	5	4.9	very good	2021-11-14	3
2	2	3.5	I donot like it	2021-10-14	4
3	4	4.8	It is good	2021-02-14	5
4	2	4.6	Keep going	2021-08-14	6

Query1 basic query

- **1. Retrieve all employee in Restaurant;**
  - select * from business;

business_id	full_name	phone_number	email	username
1	Lily Lee	1 674-823-4089	lily@gmail.com	lily
2	Jeff Davis	1 625-828-9045	jeff@gmail.com	jeff
3	Linda Smith	1 925-563-6525	linda@gmail.com	lida
4	Monica Wilson	1 925-453-9999	wilson@gmail.com	wilson

R ← σ (business)

Query2 basic query

- **2. Retrieve the rating and the menu name which score larger than 4.0.**

  - select menu_name, score, remarks, date_recorded
  - from rating, menu
  - where score > 4.0 and rating.menu_id = menu.menu_id;

menu_name	score	remarks	date_recorded
Egg Roll	4.9	very good	2021-11-14
Dandan Noodles	4.8	It is good	2021-02-14
Spicy Fish	4.6	Keep going	2021-08-14

$menu\_comment \leftarrow \sigma\ (rating) \bowtie (menu)$

$R \leftarrow \pi_{menu\_name,\ score,\ remarks,\ date\_recorded}\ (\sigma_{score\ >\ 4.0}(menu\_comment\ ))$

Query3 basic query

- **3. Retrieve all payment which process by “Jeff Davis”.**

  - select order_id, amount_$, paid_by, payment_date
  - from payment as p, business as b
  - where full_name = 'Jeff Davis ' and p.processed_by = b.business_id;

order_id	amount_$	paid_by	payment_date
3	19.9	Peter	2020-10-14
5	55.7	Chang	2020-05-12
6	39.9	Ken	2020-03-28

$$business\_payment \leftarrow \sigma\ (payment) \bowtie_{processed\_by = business\_id} (business)$$
$$R \leftarrow \pi_{order\_id,\ amount\_\$,\ paid\_by,\ payment\_date}\ (\sigma_{full\_name = 'Jeff\ Davis\ '}\ (business\_payment))$$

Query4 basic query

- **4. Retrieve total order total amount after April 2020;**

  - select round(sum(amount_$), 2) as sum_amount_$
  - from payment
  - where payment_date >= '20200401';

sum_amount_$
115.5

R (sum_amount_$) ← $\Im_{SUM}$ amount_$ (payment)

Query5 nest query

- **5. Retrieve all customer information who don't checkout.**

  - select * from customer where customer_id
  - in (select customer_id from orderInfo where order_status = 0);

customer_id	first_name	last_name	email	phone_number	username
1	Nacy	Smith	smith@gmai.com	1 888-348-9045	chen
2	William	Brown	brown@gmai.com	1 888-601-5957	brown

$cust\_status\_0 \leftarrow \pi_{customer\_id}(\sigma_{order\_status = 0}(orderInfo))$

$R \leftarrow \sigma_{full\_name = \text{'Jeff Davis'}}(customer) \bowtie cust\_status\_0$

Query6 nest query

- **6. Retrieve the names of customer who don't comment the menu;**

  - select first_name, last_name
  - from customer
  - where customer_id not in
  - (select distinct customer_id from rating);

first_name	last_name
Nacy	Smith
William	Brown
Emily	jing
John	ding

cust ← $\pi_{customer\_id}$ (customer) - $\pi_{customer\_id}$ (rating)
R ← $\pi_{first\_name,\ last\_name}$ (customer ⋈ (cust)

Query7 nest query

- **7. Retrieve the customer who do not order anything before;**

  - select first_name, last_name, email
  - from customer
  - where customer_id not in (
  - select distinct customer_id from orderInfo);

first_name	last_name	email
Emily	jing	jing@gmai.com
John	ding	ding@gmai.com

cust ← $\pi_{customer\_id}$ (customer) - $\pi_{customer\_id}$ (orderInfo)
R ← $\pi_{first\_name, last\_name, email}$ (customer ⋈ (cust)

Query8 Inner join

- **8. Retrieve all customer order and order detail information.**
  - select first_name, last_name, o.order_id, o.total_amount_$, menu_id, quantity
  - from customer c inner join orderInfo as o on c.customer_id = o.customer_id
  - inner join orderdetail as d
  - on o.order_id = d.order_id;

first_name	last_name	order_id	total_amount_$	menu_id	quantity
Nacy	Smith	1	19.9	1	1
William	Brown	2	59.7	2	1
William	Brown	2	59.7	3	2
Emily	Jones	3	19.9	5	1
Jimmy	Henry	4	39.9	2	1
James	Johnson	5	55.7	4	1
James	Johnson	5	55.7	1	1
Json	Chang	6	39.9	2	1

$cust\_order \leftarrow (customer \bowtie (orderInfo)$
$order\_detail \leftarrow (cust\_order \bowtie (orderdetail)$
$R \leftarrow \pi_{first\_name,\ last\_name,\ o.order\_id,\ o.total\_amount\_\$,\ menu\_id,\ quantity} (order\_detail)$

Query9 Out join

- **9. Retrieve business information who serve the customer, the customer last_name = ‘Henry’.**
  - select full_name, b.phone_number, b.email
  - from business b right join orderInfo as o on b.business_id = o.processed_by
  - left join customer as c
  - on o.customer_id = c.customer_id
  - where c.last_name = 'Henry’;

full_name	phone_number	email
Monica Wilson	1 925-453-9999	wilson@gmail.com

$busi\_order \leftarrow (business \bowtie_{business\_id = processed\_by} (orderInfo)$
$order\_detail \leftarrow (customer \bowtie (busi\_order)$
$R \leftarrow \pi_{full\_name, phone\_number, email} (\sigma_{last\_name = 'Henry'} (order\_detail))$

Query10 3-way join

- **10. Retrieve all customer who ordered mene_name = ‘Spicy Fish’.**
  - select first_name, last_name, email
  - from customer c
  - inner join orderInfo as o on c.customer_id = o.customer_id
  - inner join orderdetail as d on o.order_id = d.order_id
  - right join menu as m on d.menu_id = m.menu_id
  - where menu_name = 'Spicy Fish’;

first_name	last_name	email
William	Brown	brown@gmai.com
Jimmy	Henry	henry@gmai.com
Json	Chang	chang@gmai.com

cust_order ← (customer ⋈ (orderInfo)
cust_order _detail ← (orderdetail ⋈ (cust_order)
cust_menu← (menu ⋈ (cust_order _detail)
R ← π first_name, last_name, email (σ menu_name = ‘Spicy Fish’ (cust_menu))

Conclusion

- Order management system provide business a digital way to manage restaurant.
- ER diagrams analyze existing databases to find and resolve problems in logic or deployment.
- Basic query Nested query and Join.
- Algebra Relationship.

Thank you!