

NYU – TANDON SCHOOL OF ENGINEERING

CS-GY 6083B, Spring 2023

Principles of Database Systems

Group Project Part 2 Report

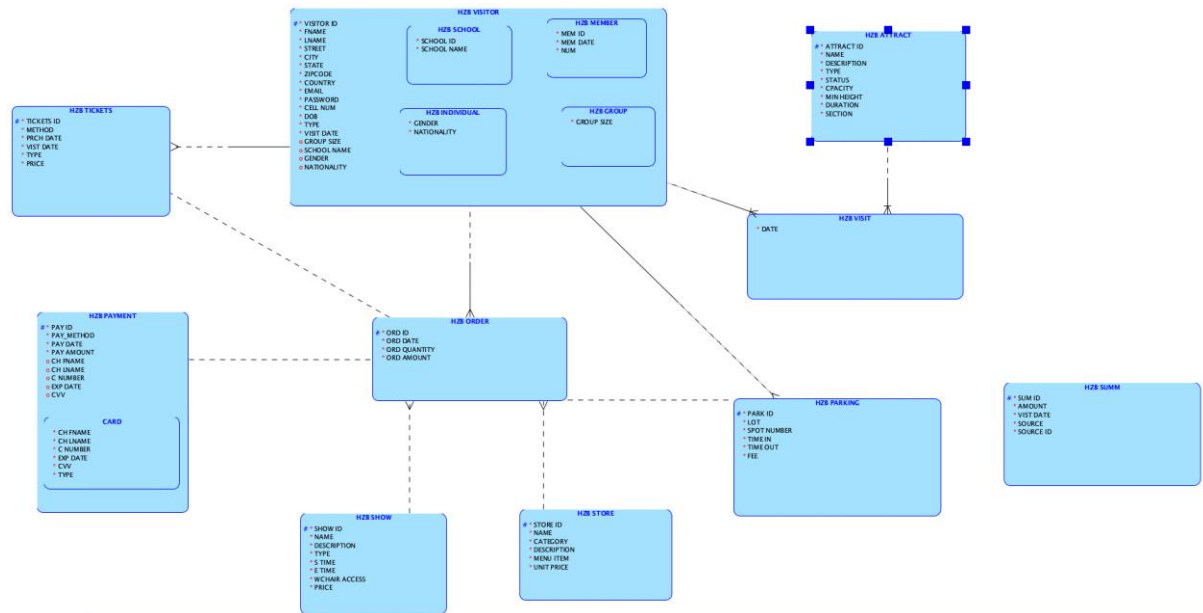
Student ID:

hq2057 zz3875 bh2612

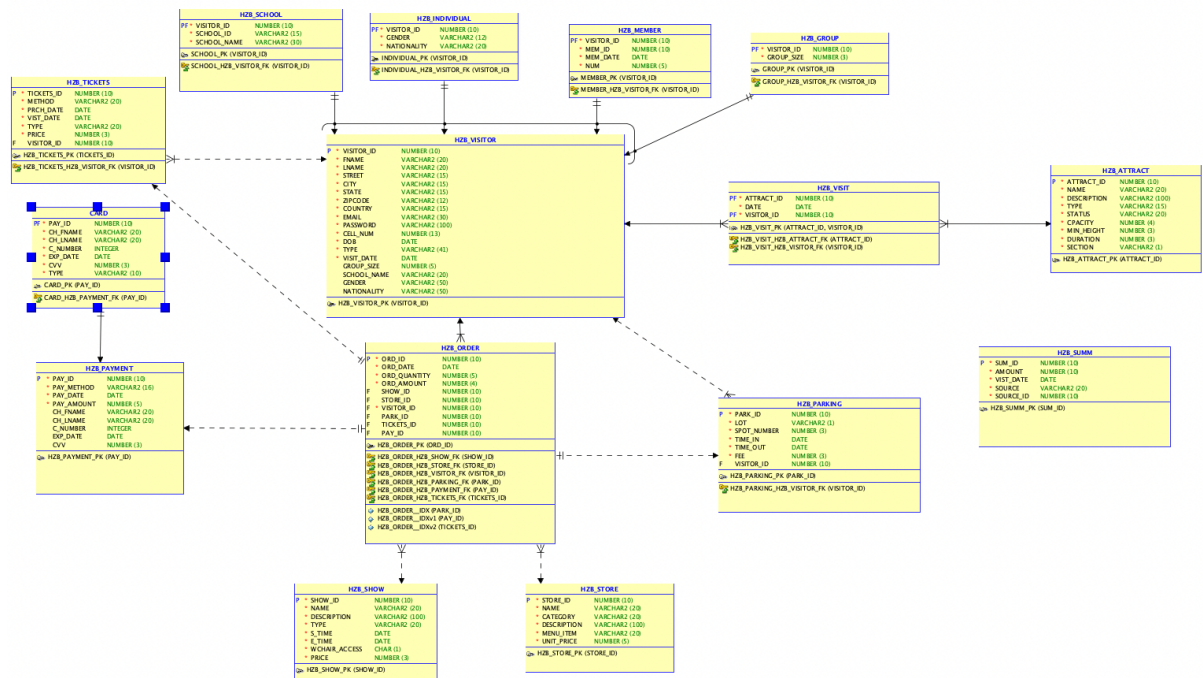
Full Name:

Hanbin Qin Zehan Zhao Baitong Hu

Logical Model



Relation Model



Execute Summary

For customers of VOA, the main function of the Web App is to purchase tickets, browse all attractions, shows, store items and parkings. All of these orders, which should have a corresponding payment, should be processed by a VOA employee. For employees of VOA, they can create, read, update, and delete the major tables in the database.

Setup instruction:

- set up MySQL database using file in /server/MySQL/MySQL.sql
- change database connection parameter in /server/src/dbUtil.js
- run npm install in both the /server and /client folder.
- use npm start in both /server and /client folder to run the server and Web App
- the Web App will by default in <http://localhost:3000>

Assumption:

1. One visitor can buy any number of tickets.
2. One visitor in the database can be saved without buying tickets today, since last visiting. So visitors in the database are optional to have a ticket.
3. One visitor can make their choice for visit attractions or not, attractions can have no visitors.
4. One visitor can have no parking, but parking needs a visitor id.
5. One visitor id is needed to buy a ticket.
6. One visitor doesn't need to make an order, but to make an order, visitor id is needed.
7. You can set the payment method without an order during the sign in process.
8. An order can make without payment, so visitors can drop their order before they make the payment.
9. Tickets, Show, Store and Parking can be in the order but not necessary, order can be placed only with visitor id.
10. Data in summary must have one source and source id associated with one of ticket, show, store, parking and must have one corresponding visitor id.
11. Order will be placed with show id or store id, since the data in show and store exists before order.
12. Buying tickets or parking will produce an order in the order database with tickets id or parking id, since the id will be produced after insert.
13. If visitors have membership, when they purchase tickets they will automatically use the membership discount.
14. Procedure input must be in format 'date-month-year' as 'dd-mmm-yy'.
15. One ticket can use multiply payment to pay all amount
16. One ticket transaction is associated with exactly one order transaction, and one order transaction is associated with exactly one ticket transaction
17. One parking transaction is associated with exactly one order transaction, and one one order transaction is associated with exactly one parking transaction.
18. One payment record is associated with exactly one order transaction, and one order transaction is associated with exactly one payment record.
19. The optional attributes in hzb_visitor and hzb_payment are for the subtype table. Users in

corresponding types need to enter the information while making payment or registering.

Brief details of software, programming language, and database used

Design	Oracle Data Modeler
Code Editor	Visual Studio Code
Database	MySQL
Backend	JavaScript, Express
Frontend	JavaScript CSS HTML, React
Server	Local

We use Javascript to implement both frontend and backend, mysql as the database management system.

We use VS Code to implement the code, mysql workbench to store the local code, and node js as the environment to run the code.

For the backend, we use Express JS to handle HTTP requests and manipulate the database.

For the frontend, we use React to implement the web application.

Screenshots of Web Application

Login

Email

Password

[Login](#)

Don't have an account? [Register](#)

Register

First Name

Last Name

Gender

Email

Password

Street

City

State

Zipcode

Country

Cell Number

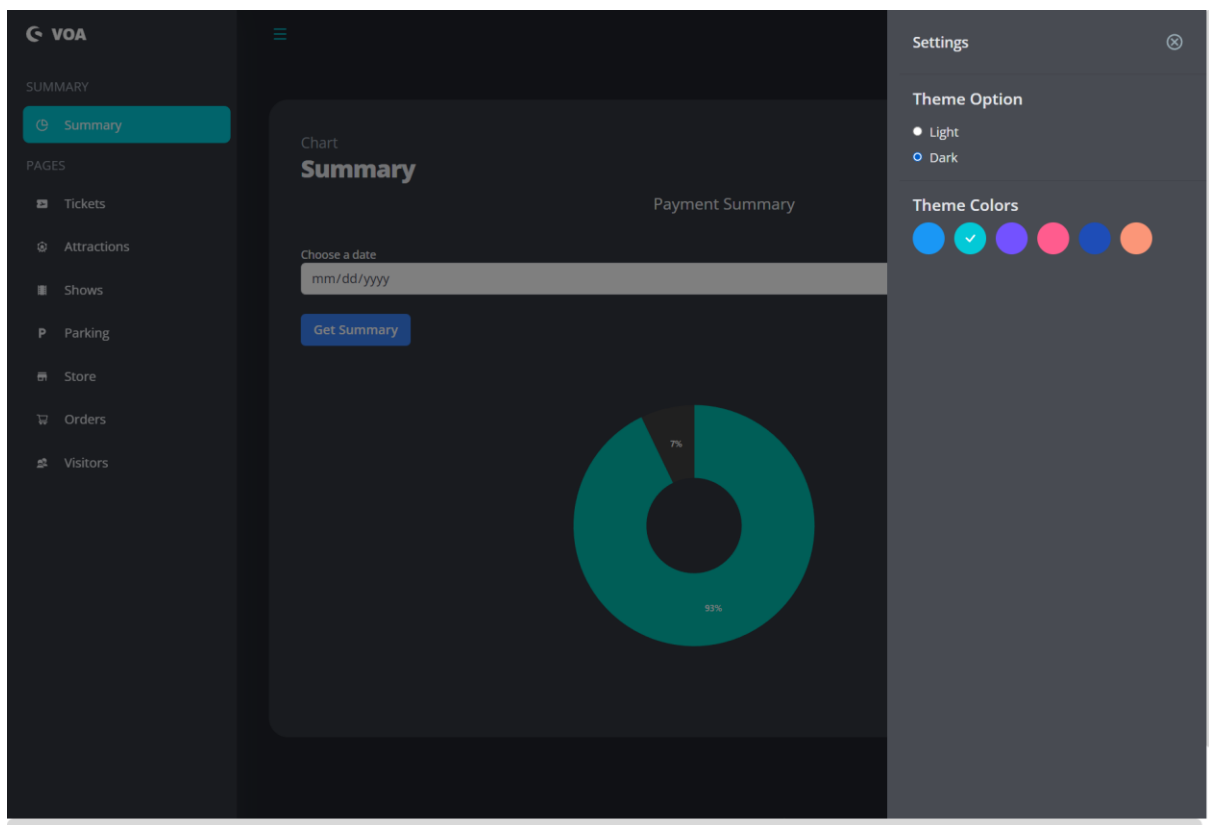
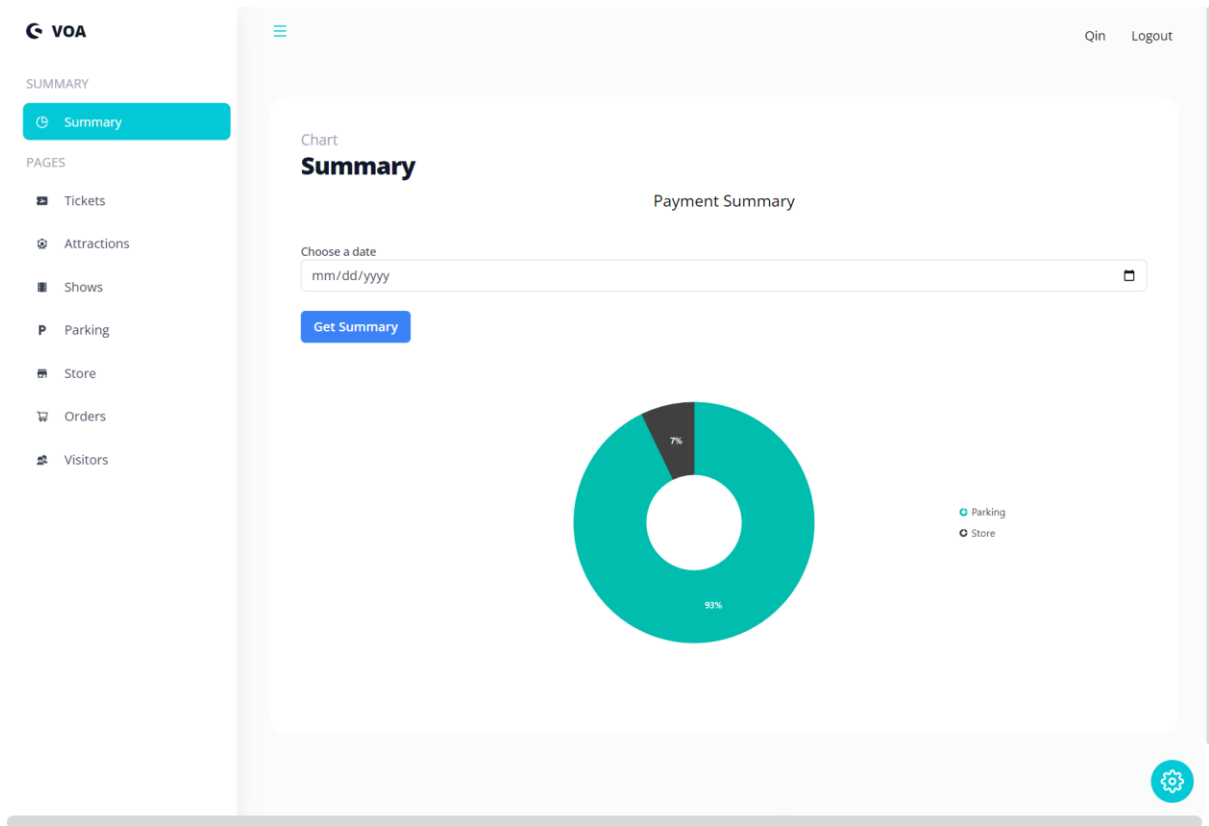
Date of Birth

Type

Visit Date

[Register](#)

Already have an account? [Login](#)



VOA

SUMMARY

Summary

PAGES

Tickets

Attractions

Shows

Parking

Store

Orders

Visitors

Page

Tickets

ID	Purchase Meth...	Purchase Date	Visit Date	Type	Price	Visitor ID
1	Online	2023-05-01T04:...	2023-06-30T04:...	Child	20	1
2	Onsite	2023-05-02T04:...	2023-06-30T04:...	Adult	35	2
3	Online	2023-05-03T04:...	2023-06-30T04:...	Child	20	3
4	Onsite	2023-05-04T04:...	2023-06-30T04:...	Senior	25	4
5	Online	2023-05-05T04:...	2023-06-30T04:...	Member	15	5
6	Onsite	2023-05-06T04:...	2023-06-30T04:...	Adult	35	6
7	Online	2023-05-07T04:...	2023-06-30T04:...	Child	20	7
8	Onsite	2023-05-08T04:...	2023-06-30T04:...	Senior	25	8
9	Online	2023-05-10T04:...	2023-06-30T04:...	Child	30	7
10	Online	2023-05-10T04:...	2023-05-10T04:...	Individual	35	11
11	Online	2023-05-10T04:...	2023-05-03T04:...	Member	35	11
12	Online	2023-05-10T04:...	2023-05-18T04:...	Group	35	11

1 of 2 pages (14 items)

CS-GY6083B Group Project

Page

Attractions

ID	Name	Description	Type	Status	Capacity	Min Height	Duration	Sect...
1	Roller Coaster 1	A thrilling roller coaster ride	roller coaster	Open	50	120	180	A
2	Water Ride 1	A fun water ride for all ages	water ride	Open	30	100	120	B
3	Dark Ride 1	A spooky dark ride with special effects	dark ride	Open	20	90	150	C
4	Kid Ride 1	A gentle ride for young kids	kid ride	Open	15	80	60	D
5	Roller Coaster 2	Another exciting roller coaster ride	roller coaster	Open	40	110	210	E
6	Water Ride 2	A thrilling water ride with slides	water ride	Open	25	95	90	F
7	Dark Ride 2	An adventurous dark ride with special effects	dark ride	Open	18	88	120	G
8	Kid Ride 2	A fun ride designed for young children	kid ride	Open	12	75	45	H
9	Roller Coaster 3	A high-speed roller coaster for thrill-seekers	roller coaster	Open	35	130	180	I
1...	Water Ride 3	A family-friendly water ride with a lazy river	water ride	Open	28	105	150	I

1 of 1 pages (10 items)

AddUpdateDelete

CS-GY6083B Group Project

VOA

SUMMARY

Summary

PAGES

Tickets

Attractions

Shows

Parking

Store

Orders

Visitors

QinLogout

Page

Update Attraction

Attraction ID : 4

Name

description

Description

type

Type

please select status

Status

capacity

Capacity

Min Height

Min Height

duration

Duration

section

section

Update Attraction

Back

VOA

SUMMARY

Summary

PAGES

Tickets

Attractions

Shows

Parking

Store

Orders

Visitors

QinLogout

Page

Update Store

Store ID : 4

Name

category

please select type

Description

description

Menu Item

menu item

Unit Price

unit price

Update Store

Back

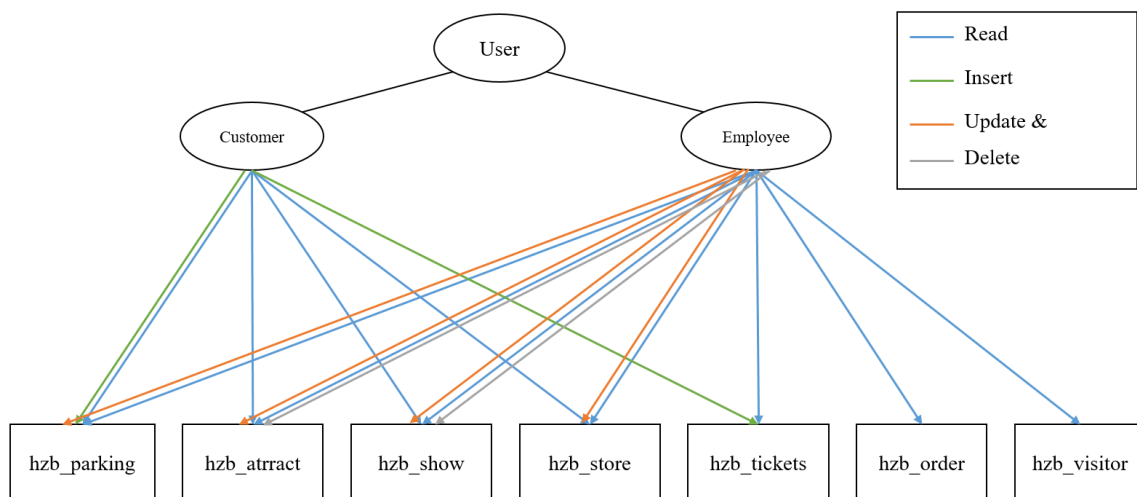
CS-GY6083B Group Project

Security Features

1. **SQL Injection:** SQL Injection is a serious problem that can occur when an attacker inputs data that includes code injection, which directly manipulates the database. However, our web application is protected against this type of attack by using parameterized queries to manipulate databases. For example, when inserting data into the `hzb_attract` table, we use the following query:

```
INSERT INTO hzb_attract (name, description, type, status, capacity, min_height, duration, section) VALUES (?, ?, ?, ?, ?, ?, ?, ?)
```

To fill each “?” placeholder with the value in the request, we pass the actual values as a separate array to the query function. This ensures that input values are properly sanitized and prevents SQL Injection attacks from occurring.
2. **Cross Site Scripting:** We use React to build out our entire web application, as it provides XSS protection and prevents most XSS attacks. React automatically escapes user input before rendering, ensuring that any potentially harmful code is neutralized. Additionally, we pass a function instead of code directly when adding an event handler, further protecting against potential security vulnerabilities. Furthermore, we ensure that we do not include any JavaScript code inside content, further minimizing the risk of XSS attacks.
3. The password is encrypted and stored in database.
4. **User authority on Web interface:** In frontend Web app, the users are divided into two groups: Employee and Customer. Different types of users have different access to the functionality. For example, employees can perform Insert, Update and Delete operations on tables including Attraction, Show, Store and Parking. Customers can only read the records on tables including Attraction, Show, Store and Parking. The detailed data accessibility are show as follow diagram:



Reflection

When creating a project that uses MySQL as a database, it is important to carefully map out your data structure and relationships. This may involve creating an ER diagram or using a tool like MySQL Workbench to visualize your database schema.

Once you have your database structure mapped out, you can start building your backend with Express. This includes creating routes to handle incoming HTTP requests and performing CRUD (Create, Read, Update, Delete) operations on the database. You will also need to implement authentication and authorization to ensure that only authorized users can access certain routes and data.

With the back-end in place, you can move on to building the front-end with React Router. This involves creating a set of routes that correspond to different views or pages in your application and using React components to render the appropriate content for each route.

Overall, building a project with MySQL, Express, and React Router can be a challenging but rewarding experience. By carefully planning your database structure, building a robust backend with Express, and creating a responsive frontend with React Router, you can create a robust, scalable application that meets the needs of your users.

Time management is a constraint for us and we should start early and start often.

List of tables

select * from hzb_attract

	attract_id	name	description	type	status	capacity	min_height	duration	section
▶	1	Roller Coaster 1	A thrilling roller coaster ride	roller coaster	Open	50	120	180	A
	2	Water Ride 1	A fun water ride for all ages	water ride	Open	30	100	120	B
	3	Dark Ride 1	A spooky dark ride with special effects	dark ride	Open	20	90	150	C
	4	Kid Ride 1	A gentle ride for young kids	kid ride	Open	15	80	60	D
	5	Roller Coaster 2	Another exciting roller coaster ride	roller coaster	Open	40	110	210	E
	6	Water Ride 2	A thrilling water ride with slides	water ride	Open	25	95	90	F
	7	Dark Ride 2	An adventurous dark ride with special effects	dark ride	Open	18	88	120	G
	8	Kid Ride 2	A fun ride designed for young children	kid ride	Open	12	75	45	H
	9	Roller Coaster 3	A high-speed roller coaster for thrill-seekers	roller coaster	Open	35	130	180	I
	10	Water Ride 3	A family-friendly water ride with a lazy river	water ride	Open	28	105	150	I
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

select * from hzb_visitor

	visitor_id	frame	lname	street	city	state	zipcode	country	email	password	cell_num	dob	type	visit_date	group_size	school_name	gender	nationality	role
▶	1	John	Doe	123 Main St	Anytown	CA	12345	USA	john.doe@email.com	password123	1234567890	1990-01-01	Member	2022-05-05	10	100	Male	United States	Customer
	2	Jane	Doe	456 Oak St	Sometown	CA	67890	USA	janedoe@email.com	password456	987654321	1995-07-15	Individual	2022-05-05	10	100	Female	United States	Customer
	3	Alex	Smith	789 Maple Ave	Anytown	CA	12345	USA	alexsmith@email.com	password789	1112223333	2005-09-01	School	2022-05-05	10	100	Male	United States	Customer
	4	Sam	Jones	321 Pine St	Other town	CA	54321	USA	samjones@email.com	passwordabc	4445556666	1998-03-22	Group	2022-05-05	20	100	Male	United States	Customer
	5	Mike	Brown	654 Elm St	Sometown	CA	67890	USA	mikebrown@email.com	passwordxyz	7778889999	1992-12-01	Member	2022-05-05	10	100	Male	United States	Customer
	6	Karen	Lee	987 Oak Ave	Other town	CA	54321	USA	karenlee@email.com	password123	2223334444	1985-05-11	Individual	2022-05-05	10	100	Female	United States	Customer
	7	Mark	Davis	456 Pine St	Anytown	CA	12345	USA	markdavis@email.com	password456	5556667777	2007-10-15	School	2022-05-05	10	100	Male	United States	Customer
	8	Mary	Wong	567 Camp St	Anytown	CA	12345	USA	marywong@email.com	password567	5556667777	2000-04-26	School	2022-05-06	10	100	Female	United States	Customer
	9	Ming	Ming	678 Peter St	Anytown	CA	12345	USA	mingming@email.com	password123	5556667777	2000-08-11	School	2022-05-06	10	100	Male	United States	Customer
	10	Peter	Pan	123 Pine St	Anytown	CA	12345	USA	peterpan@email.com	password345	5556667777	2001-09-15	School	2022-05-06	10	100	Male	United States	Customer
	11	Jack	Wong	421 Jackson St	Sometown	CA	12345	USA	jackwong@email.com	passwordabc	8889990000	1998-01-30	Group	2022-05-06	15	100	Male	United States	Customer
	12	Emily	Kim	321 Elm St	Sometown	CA	67890	USA	emilykim@email.com	passwordabc	8889990000	1999-06-30	Group	2022-05-05	15	100	Female	United States	Customer
	13	Mat	Meng	789 park St	Sometown	CA	23456	USA	matmeng@email.com	passwordabc	8889990000	1992-04-10	Group	2022-05-05	4	100	Male	United States	Customer
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

select * from hzb_group

	visitor_id	group_size
▶	4	20
	11	15
	12	15
	13	4
*	NULL	NULL

select * from hzb_individual

	visitor_id	gender	nationality
▶	2	Male	United States
	6	Male	United States
*	NULL	NULL	NULL

select * from hzb_member

	visitor_id	mem_id	mem_date	num
▶	1	1	2022-05-05	4
	5	5	2022-05-05	4
*	NULL	NULL	NULL	NULL

select * from hzb_school

	visitor_id	school_id	school_name
▶	3	3	River
	7	7	South
	8	8	South
	9	9	South
	10	10	South
*	NULL	NULL	NULL

select * from hzb_order

	ord_id	ord_date	ord_quantity	ord_amount	show_id	store_id	visitor_id	park_id	tickets_id	pay_id
▶	1	2023-06-30	1	17	NULL	NULL	1	NULL	1	NULL
	2	2023-06-30	1	35	NULL	NULL	2	NULL	2	NULL
	3	2023-06-30	1	19	NULL	NULL	3	NULL	3	NULL
	4	2023-06-30	20	500	NULL	NULL	4	NULL	4	NULL
	5	2023-06-30	1	13	NULL	NULL	5	NULL	5	NULL
	6	2023-06-30	1	35	NULL	NULL	6	NULL	6	NULL
	7	2023-06-30	1	19	NULL	NULL	7	NULL	7	NULL
	8	2023-06-30	1	25	NULL	NULL	8	NULL	8	NULL
	9	2023-06-30	1	29	NULL	NULL	7	NULL	9	NULL
	10	2023-05-01	2	50	NULL	NULL	1	1	NULL	NULL
	11	2023-05-01	2	50	NULL	NULL	2	2	NULL	NULL
	12	2023-05-01	2	50	NULL	NULL	3	3	NULL	NULL
	13	2023-05-01	2	50	NULL	NULL	4	4	NULL	NULL
	14	2023-05-01	2	50	NULL	NULL	5	5	NULL	NULL
	15	2023-05-01	2	50	NULL	NULL	6	6	NULL	NULL
	16	2023-05-01	2	50	NULL	NULL	7	7	NULL	NULL
	17	2023-05-01	2	50	NULL	NULL	8	8	NULL	NULL
	18	2023-05-01	2	50	NULL	NULL	7	9	NULL	NULL
	19	2023-05-01	2	50	NULL	NULL	6	10	NULL	NULL
	20	2023-05-01	2	50	NULL	NULL	5	11	NULL	NULL
	21	2023-05-01	2	100	NULL	NULL	4	12	NULL	NULL
	22	2023-05-01	2	150	NULL	NULL	3	13	NULL	NULL
	23	2023-04-04	1	11	1	NULL	1	NULL	NULL	NULL
	24	2023-04-06	1	8	2	NULL	2	NULL	NULL	NULL
	25	2023-04-07	1	15	3	NULL	3	NULL	NULL	NULL
	26	2023-04-08	1	10	4	NULL	4	NULL	NULL	NULL
	27	2023-04-09	1	12	5	NULL	5	NULL	NULL	NULL
	28	2023-04-11	1	8	6	NULL	6	NULL	NULL	NULL
	29	2023-04-12	1	12	7	NULL	7	NULL	NULL	NULL
	30	2023-04-13	1	15	8	NULL	8	NULL	NULL	NULL
	31	2023-04-14	1	25	9	NULL	1	NULL	NULL	NULL
	32	2023-04-10	1	13	10	NULL	2	NULL	NULL	NULL
	33	2023-04-04	1	11	1	NULL	3	NULL	NULL	NULL
	34	2023-04-06	1	8	2	NULL	4	NULL	NULL	NULL
	35	2023-04-07	1	15	3	NULL	5	NULL	NULL	NULL

select * from hzb_parking

	park_id	lot	spot_number	time_in	time_out	fee	visitor_id
▶	1	A	33	2023-05-01	2023-05-01	50	1
	2	B	34	2023-05-01	2023-05-01	50	2
	3	C	35	2023-05-01	2023-05-01	50	3
	4	D	36	2023-05-01	2023-05-01	50	4
	5	E	37	2023-05-01	2023-05-01	50	5
	6	F	38	2023-05-01	2023-05-01	50	6
	7	G	39	2023-05-01	2023-05-01	50	7
	8	H	40	2023-05-01	2023-05-01	50	8
	9	I	41	2023-05-01	2023-05-01	50	7
	10	J	42	2023-05-01	2023-05-01	50	6
	11	A	43	2023-05-01	2023-05-01	50	5
	12	B	44	2023-05-01	2023-05-01	100	4
	13	C	45	2023-05-01	2023-05-01	150	3
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

select * from hzb_store

	store_id	name	category	description	menu_item	unit_price
▶	1	Food Stall 2	Food Stall	This is another food stall selling snacks and bev...	Fries	3
	2	Ice Cream Parlor 1	Ice Cream Parlor	This is an ice cream parlor offering a wide range...	Chocolate Sundae	7
	3	Ice Cream Parlor 2	Ice Cream Parlor	This is another ice cream parlor with unique ice ...	Strawberry Cone	4
	4	Restaurant 1	Restaurant	This is a restaurant serving fine dining meals wit...	Grilled Salmon	15
	5	Restaurant 2	Restaurant	This is another restaurant specializing in Italian ...	Margherita Pizza	10
	6	Gift Shop 1	Gift Shop	This is a gift shop offering souvenirs, toys, and ...	Keychain	2
	7	Gift Shop 2	Gift Shop	This is another gift shop with a wide range of co...	T-shirt	12
	8	Apparel Store 1	Apparels	This is an apparel store selling trendy clothing f...	Jeans	30
	9	Apparel Store 2	Apparels	This is another apparel store offering a variety ...	Sunglasses	20
	10	Food Stall 3	Food Stall	This is a new food stall offering exotic cuisines f...	Sushi Roll	8
	11	Ice Cream Parlor 3	Ice Cream Parlor	This is a popular ice cream parlor known for its u...	M Ice Cream	6
	12	Gift Shop 3	Gift Shop	This is a small gift shop offering local crafts and ...	Handmade Pottery	18
*	NULL	NULL	NULL	NULL	NULL	NULL

select * from hzb_tickets

	tickets_id	method	prch_date	vist_date	type	price	visitor_id
▶	1	Online	2023-05-01	2023-06-30	Child	20	1
	2	Onsite	2023-05-02	2023-06-30	Adult	35	2
	3	Online	2023-05-03	2023-06-30	Child	20	3
	4	Onsite	2023-05-04	2023-06-30	Senior	25	4
	5	Online	2023-05-05	2023-06-30	Member	15	5
	6	Onsite	2023-05-06	2023-06-30	Adult	35	6
	7	Online	2023-05-07	2023-06-30	Child	20	7
	8	Onsite	2023-05-08	2023-06-30	Senior	25	8
	9	Online	2023-05-10	2023-06-30	Child	30	7
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

select * from hzb_visitor

	visitor_id	fname	lname	street	city	state	zipcode	country	email	password	cell_num	dob	type	visit_date	group_size	school_name	gender	nationality	role
▶	1	John	Doe	123 Main St	Anytown	CA	12345	USA	john.doe@email.com	password123	1234567890	1990-01-01	Member	2022-05-05	10	NULL	Male	United States	Customer
	2	Jane	Doe	456 Oak St	Sometown	CA	67890	USA	jane.doe@email.com	password456	987654321	1995-07-15	Individual	2022-05-05	10	NULL	Male	United States	Customer
	3	Alex	Smith	789 Maple Ave	Anytown	CA	12345	USA	alex.smith@email.com	password789	1112223333	2005-09-01	School	2022-05-05	10	NULL	Male	United States	Customer
	4	Sam	Jones	321 Pine St	Other town	CA	54321	USA	sam.jones@email.com	passwordabc	4445556666	1998-03-22	Group	2022-05-05	20	NULL	Male	United States	Customer
	5	Mike	Brown	654 Elm St	Sometown	CA	67890	USA	mike.brown@email.com	passwordxyz	7778889999	1992-12-01	Member	2022-05-05	10	NULL	Male	United States	Customer
	6	Karen	Lee	987 Oak Ave	Other town	CA	54321	USA	karen.lee@email.com	password123	2223334444	1985-05-11	Individual	2022-05-05	10	NULL	Male	United States	Customer
	7	Mark	Davis	456 Pine St	Anytown	CA	12345	USA	mark.davis@email.com	password456	5556667777	2007-10-15	School	2022-05-05	10	South	Male	United States	Customer
	8	Mary	Wong	567 Camp St	Anytown	CA	12345	USA	mary.wong@email.com	password567	5556667777	2000-04-26	School	2022-05-06	10	South	Male	United States	Customer
	9	Ming	Ming	678 Peter St	Anytown	CA	12345	USA	ming.ming@email.com	password123	5556667777	2000-08-11	School	2022-05-06	10	South	Male	United States	Customer
	10	Peter	Pan	123 Pine St	Anytown	CA	12345	USA	peter.pan@email.com	password345	5556667777	2001-09-15	School	2022-05-06	10	South	Male	United States	Customer
	11	Jack	Wong	421 Jackson St	Sometown	CA	12345	USA	jack.wong@email.com	passwordabc	8889990000	1998-01-30	Group	2022-05-06	15	NULL	Male	United States	Customer
	12	Emily	Kim	321 Elm St	Sometown	CA	67890	USA	emily.kim@email.com	passwordabc	8889990000	1999-06-30	Group	2022-05-05	15	NULL	Male	United States	Customer
	13	Mat	Meng	789 park St	Sometown	CA	23456	USA	mat.meng@email.com	passwordabc	8889990000	1992-04-10	Group	2022-05-05	4	NULL	Male	United States	Customer
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

select * from hzb_payment

	pay_id	pay_method	pay_date	pay_amount	ch_fname	ch_lname	c_number	exp_date	cvv	type
▶	1	Cash	2023-04-06	201	NULL	NULL	NULL	NULL	NULL	NULL
	2	Credit	2023-04-06	500	John	Doe	1234213124143	2024-12-31	123	NULL
	3	Debit	2023-04-06	151	Jane	Smith	9876543210987654	2023-06-30	456	NULL
	4	Cash	2023-04-07	100	NULL	NULL	NULL	NULL	NULL	NULL
	5	Credit	2023-04-07	300	Mark	Johnson	7890123456789012	2025-08-31	789	NULL
	6	Debit	2023-04-08	50	Sarah	Wilson	5432109876543210	2023-09-30	321	NULL
	7	Cash	2023-04-08	76	NULL	NULL	NULL	NULL	NULL	NULL
	8	Credit	2023-04-09	600	David	Brown	4567890123456789	2024-10-31	567	NULL
	9	Debit	2023-04-09	251	Emily	Davis	2109876543210987	2023-07-31	234	NULL
	10	Cash	2023-04-10	50	NULL	NULL	NULL	NULL	NULL	NULL
	11	Credit	2023-04-11	151	Michael	Taylor	2109876543210987	2023-07-31	234	NULL

select * from hzb_card

	pay_id	ch_fname	ch_lname	c_number	exp_date	cvv	type
▶	2	John	Doe	1234213124143	2024-12-31	123	Credit
	3	Jane	Smith	9876543210987654	2023-06-30	456	Debit
	5	Mark	Johnson	7890123456789012	2025-08-31	789	Credit
	6	Sarah	Wilson	5432109876543210	2023-09-30	321	Debit
	8	David	Brown	4567890123456789	2024-10-31	567	Credit
	9	Emily	Davis	2109876543210987	2023-07-31	234	Debit
	11	Michael	Taylor	2109876543210987	2023-07-31	234	Credit

select * from hzb_visit

	attract_id	DATE	visitor_id
	1	2023-04-04	1
	1	2023-04-04	2
	1	2023-04-04	3
	1	2023-04-04	4
	2	2023-04-03	1
	2	2023-04-03	2
	2	2023-04-03	3
	2	2023-04-03	5
▶	2	2023-04-03	7
	2	2023-04-03	8
	3	2023-04-02	2
	3	2023-04-02	3
	3	2023-04-02	4
	3	2023-04-02	5
	3	2023-04-02	7
	3	2023-04-02	8
	7	2023-03-28	5
	7	2023-03-28	6

List of table and table row

	TABLE_NAME
▶	hzb_attract
	hzb_card
	hzb_group
	hzb_individual
	hzb_member
	hzb_order
	hzb_parking
	hzb_payment
	hzb_school
	hzb_show
	hzb_store
	hzb_summ
	hzb_tickets
	hzb_visit
	hzb_visitor
	hzb_attract_count
▶	10

	hzb_card_count
▶	7
	hzb_group_count
▶	4
	hzb_individual_count
▶	2
	hzb_member_count
▶	2
	hzb_order_count
▶	52
	hzb_parking_count
▶	13
	hzb_payment_count
▶	11
	hzb_school_count
▶	5
	hzb_show_count
▶	10
	hzb_store_count
▶	12
	hzb_summ_count
▶	14
	hzb_tickets_count
▶	9
	hzb_visitor_count
▶	13
	hzb_visit_count
▶	18

DDL Code

```

DROP DATABASE IF EXISTS voa_database;
CREATE DATABASE voa_database;
USE voa_database;

CREATE TABLE hzb_card (
    pay_id INT(10) NOT NULL,
    ch_fname VARCHAR(20) NOT NULL,
    ch_lname VARCHAR(20) NOT NULL,
    c_number BIGINT(50) NOT NULL,
    exp_date DATE NOT NULL,
    cvv INT(3) NOT NULL,
    type VARCHAR(20) NOT NULL
);
ALTER TABLE hzb_card
ADD CONSTRAINT hzb_card_pk PRIMARY KEY (pay_id);
ALTER TABLE hzb_card
MODIFY pay_id INT(10) NOT NULL COMMENT 'Cash, credit or debit card',
MODIFY ch_fname VARCHAR(20) NOT NULL COMMENT 'card host first name',
MODIFY ch_lname VARCHAR(20) NOT NULL COMMENT 'card host last name',
MODIFY c_number VARCHAR(20) NOT NULL COMMENT 'card number',
MODIFY exp_date DATE NOT NULL COMMENT 'card expiration date',
MODIFY cvv VARCHAR(3) NOT NULL COMMENT 'card CVV three digit number',
MODIFY type VARCHAR(10) NOT NULL COMMENT 'card type, debit or credit';
CREATE TABLE hzb_attract (
    attract_id INT(10) NOT NULL AUTO_INCREMENT,
    name VARCHAR(20) NOT NULL,
    description VARCHAR(100) NOT NULL,
    type VARCHAR(15) NOT NULL,
    status VARCHAR(20) NOT NULL,
    capacity INT(4) NOT NULL,
    min_height INT(3) NOT NULL,
    duration INT(3) NOT NULL,
    section VARCHAR(1) NOT NULL,
    PRIMARY KEY (attract_id)
);
ALTER TABLE hzb_attract
MODIFY name VARCHAR(20) NOT NULL COMMENT 'attraction name',
MODIFY description VARCHAR(100) NOT NULL COMMENT 'attraction
description',
MODIFY type VARCHAR(15) NOT NULL COMMENT 'attraction type',
MODIFY status VARCHAR(20) NOT NULL COMMENT 'attraction status, open,
closed, under maintenance',
MODIFY capacity INT(4) NOT NULL COMMENT 'attraction audience capacity',
MODIFY min_height INT(3) NOT NULL COMMENT 'visitor minimum height that
can attend attraction in cm.',
MODIFY duration INT(3) NOT NULL COMMENT 'attraction duration in

```



```

minutes',
    MODIFY section VARCHAR(1) NOT NULL COMMENT 'attraction section';
CREATE TABLE hzb_group (
    visitor_id INT(10) NOT NULL,
    group_size INT(3) NOT NULL,
    PRIMARY KEY (visitor_id)
);
ALTER TABLE hzb_group
MODIFY visitor_id INT(10) NOT NULL COMMENT '',
MODIFY group_size INT(3) NOT NULL COMMENT '';
CREATE TABLE hzb_individual (
    visitor_id INT(10) NOT NULL,
    gender VARCHAR(12) NOT NULL,
    nationality VARCHAR(20) NOT NULL,
    PRIMARY KEY (visitor_id)
);
ALTER TABLE hzb_individual
MODIFY visitor_id INT(10) NOT NULL COMMENT '',
MODIFY gender VARCHAR(12) NOT NULL COMMENT '',
MODIFY nationality VARCHAR(20) NOT NULL COMMENT '';
CREATE TABLE hzb_member (
    visitor_id INT(10) NOT NULL,
    mem_id INT(10) NOT NULL,
    mem_date DATE NOT NULL,
    num INT(5) NOT NULL,
    PRIMARY KEY (visitor_id)
);
ALTER TABLE hzb_member
MODIFY visitor_id INT(10) NOT NULL COMMENT '',
MODIFY mem_id INT(10) NOT NULL COMMENT 'rest of number of discount
ticket this member',
MODIFY mem_date DATE NOT NULL COMMENT '',
MODIFY num INT(5) NOT NULL COMMENT '';
CREATE TABLE hzb_order (
    ord_id INT(10) NOT NULL AUTO_INCREMENT,
    ord_date DATE NOT NULL,
    ord_quantity INT(5) NOT NULL,
    ord_amount INT(4) NOT NULL,
    show_id INT(10),
    store_id INT(10),
    visitor_id INT(10) NOT NULL,
    park_id INT(10),
    tickets_id INT(10),
    pay_id INT(10),
    PRIMARY KEY (ord_id)
);

```

```

ALTER TABLE hzb_order
MODIFY ord_date DATE NOT NULL COMMENT 'order date',
  MODIFY ord_quantity INT(5) NOT NULL COMMENT 'quantity of item in
order',
  MODIFY ord_amount INT(4) NOT NULL COMMENT 'amount of money of this
order',
  MODIFY show_id INT(10) COMMENT '',
  MODIFY store_id INT(10) COMMENT '',
  MODIFY visitor_id INT(10) NOT NULL COMMENT '',
  MODIFY park_id INT(10) COMMENT '',
  MODIFY tickets_id INT(10) COMMENT '',
  MODIFY pay_id INT(10) COMMENT '';
CREATE UNIQUE INDEX hzb_order__idx ON hzb_order (park_id ASC);
CREATE UNIQUE INDEX hzb_order__idxv1 ON hzb_order (pay_id ASC);
CREATE UNIQUE INDEX hzb_order__idxv2 ON hzb_order (tickets_id ASC);
CREATE TABLE hzb_parking (
  park_id INT(10) NOT NULL AUTO_INCREMENT,
  lot VARCHAR(1) NOT NULL,
  spot_number INT(3) NOT NULL,
  time_in DATE NOT NULL,
  time_out DATE NOT NULL,
  fee INT(3) NOT NULL,
  visitor_id INT(10),
  PRIMARY KEY (park_id)
);
ALTER TABLE hzb_parking
MODIFY lot VARCHAR(1) NOT NULL COMMENT 'parking lot number',
  MODIFY spot_number INT(3) NOT NULL COMMENT 'spot number of this
parking position',
  MODIFY time_in DATE NOT NULL COMMENT 'time start parking in this
parking lot',
  MODIFY time_out DATE NOT NULL COMMENT 'time leave parking lot',
  MODIFY fee INT(3) NOT NULL COMMENT 'parking fee',
  MODIFY visitor_id INT(10) COMMENT '';
CREATE TABLE hzb_payment (
  pay_id INT(10) NOT NULL AUTO_INCREMENT,
  pay_method VARCHAR(16) NOT NULL,
  pay_date DATE NOT NULL,
  pay_amount INT(5) NOT NULL,
  ch_fname VARCHAR(20),
  ch_lname VARCHAR(20),
  c_number BIGINT(50),
  exp_date DATE,
  cvv INT(3),
  type VARCHAR(20),
  PRIMARY KEY (pay_id),

```

```

    CHECK (pay_method IN ('Cash', 'Credit', 'Debit'))
);
ALTER TABLE hzb_payment
MODIFY pay_method VARCHAR(16) NOT NULL COMMENT 'Cash, credit or debit
card',
    MODIFY pay_date DATE NOT NULL COMMENT 'payment date',
    MODIFY pay_amount INT(5) NOT NULL COMMENT 'payment amount';
CREATE TABLE hzb_school (
    visitor_id INT(10) NOT NULL,
    school_id VARCHAR(15) NOT NULL,
    school_name VARCHAR(30) NOT NULL,
    PRIMARY KEY (visitor_id)
);
ALTER TABLE hzb_school
MODIFY school_id VARCHAR(15) NOT NULL COMMENT '',
    MODIFY school_name VARCHAR(30) NOT NULL COMMENT '';
CREATE TABLE hzb_show (
    show_id INT(10) NOT NULL AUTO_INCREMENT,
    name VARCHAR(20) NOT NULL,
    description VARCHAR(100) NOT NULL,
    type VARCHAR(20) NOT NULL,
    s_time DATE NOT NULL,
    e_time DATE NOT NULL,
    wchair_access CHAR(1) NOT NULL,
    price INT(3) NOT NULL,
    PRIMARY KEY (show_id)
);
ALTER TABLE hzb_show
MODIFY name VARCHAR(20) NOT NULL COMMENT 'show name',
    MODIFY description VARCHAR(100) NOT NULL COMMENT 'show description',
    MODIFY type VARCHAR(20) NOT NULL COMMENT 'show type, drama, musical,
comedy, horror, adventure',
    MODIFY s_time DATE NOT NULL COMMENT 'start time of the show',
    MODIFY e_time DATE NOT NULL COMMENT 'End time of the show',
    MODIFY wchair_access CHAR(1) NOT NULL COMMENT 'Wheelchair
Accessibility yes or no',
    MODIFY price INT(3) NOT NULL COMMENT 'show price';
CREATE TABLE hzb_store (
    store_id INT(10) NOT NULL AUTO_INCREMENT,
    name VARCHAR(20) NOT NULL,
    category VARCHAR(20) NOT NULL,
    description VARCHAR(100) NOT NULL,
    menu_item VARCHAR(20) NOT NULL,
    unit_price INT(5) NOT NULL,
    PRIMARY KEY (store_id)
);

```

```

ALTER TABLE hzb_store
MODIFY name VARCHAR(20) NOT NULL COMMENT 'store name',
    MODIFY category VARCHAR(20) NOT NULL COMMENT 'store category Food
stall, Ice cream parlor, Restaurant, Gift Shop, Apparels',
    MODIFY description VARCHAR(100) NOT NULL COMMENT 'store description',
    MODIFY menu_item VARCHAR(20) NOT NULL COMMENT 'item in this store
menu',
    MODIFY unit_price INT(5) NOT NULL COMMENT 'item unit price';
CREATE TABLE hzb_summ (
    sum_id INT(10) NOT NULL AUTO_INCREMENT,
    amount INT(10) NOT NULL,
    vist_date DATE NOT NULL,
    source VARCHAR(20) NOT NULL,
    source_id INT(10) NOT NULL,
    visitor_id INT(10) NOT NULL,
    PRIMARY KEY (sum_id)
);
ALTER TABLE hzb_summ
MODIFY amount INT(10) NOT NULL COMMENT 'amount of money',
    MODIFY vist_date DATE NOT NULL COMMENT 'visit date of this summary',
    MODIFY source VARCHAR(20) NOT NULL COMMENT 'source of this payment
show, store, parking, ticket',
    MODIFY source_id INT(10) NOT NULL COMMENT 'source id of this
payment.';
CREATE TABLE hzb_tickets (
    tickets_id INT(10) NOT NULL AUTO_INCREMENT,
    method VARCHAR(20) NOT NULL,
    prch_date DATE NOT NULL,
    vist_date DATE NOT NULL,
    type VARCHAR(20) NOT NULL,
    price INT(3) NOT NULL,
    visitor_id INT(10),
    PRIMARY KEY (tickets_id)
);
ALTER TABLE hzb_tickets
MODIFY method VARCHAR(20) NOT NULL COMMENT 'Online or onsite',
    MODIFY prch_date DATE NOT NULL COMMENT 'Purchase date of the ticket',
    MODIFY vist_date DATE NOT NULL COMMENT 'Visit date of the ticket',
    MODIFY type VARCHAR(20) NOT NULL COMMENT 'Child, adult, senior or
member.',
    MODIFY price INT(3) NOT NULL COMMENT 'tickets price';
CREATE TABLE hzb_visit (
    attract_id INT(10) NOT NULL,
    `DATE` DATE NOT NULL,
    visitor_id INT(10) NOT NULL,
    PRIMARY KEY (attract_id, visitor_id)

```

```

);
ALTER TABLE hzb_visit
MODIFY `DATE` DATE NOT NULL COMMENT 'date visitor attend attraction';
CREATE TABLE hzb_visitor (
    visitor_id INT(10) NOT NULL AUTO_INCREMENT,
    fname VARCHAR(20) NOT NULL COMMENT 'visitor first name',
    lname VARCHAR(20) NOT NULL COMMENT 'visitor last name',
    street VARCHAR(15) NOT NULL COMMENT 'visitor address',
    city VARCHAR(15) NOT NULL,
    state VARCHAR(15) NOT NULL,
    zipcode VARCHAR(12) NOT NULL,
    country VARCHAR(15) NOT NULL,
    email VARCHAR(30) NOT NULL COMMENT 'visitor email',
    password VARCHAR(100) NOT NULL,
    cell_num BIGINT(13) NOT NULL COMMENT 'visitor cell number',
    dob DATE NOT NULL COMMENT 'visitor age',
    type VARCHAR(41) NOT NULL COMMENT 'visitor type, individual, group,
member, student.',
    visit_date DATE NOT NULL COMMENT 'Last time visit',
    group_size INT(5),
    school_name VARCHAR(20),
    gender VARCHAR(10),
    nationality VARCHAR(15),
    PRIMARY KEY (visitor_id)
);
ALTER TABLE hzb_visitor
ADD CONSTRAINT ch_inh_hzb_visitor CHECK (
    type IN ('Individual', 'Group', 'School', 'Member')
);
ALTER TABLE hzb_card
ADD CONSTRAINT card_hzb_payment_fk FOREIGN KEY (pay_id) REFERENCES
hzb_payment (pay_id);
ALTER TABLE hzb_group
ADD CONSTRAINT group_hzb_visitor_fk FOREIGN KEY (visitor_id) REFERENCES
hzb_visitor (visitor_id);
ALTER TABLE hzb_order
ADD CONSTRAINT hzb_order_hzb_parking_fk FOREIGN KEY (park_id) REFERENCES
hzb_parking (park_id);
ALTER TABLE hzb_order
ADD CONSTRAINT hzb_order_hzb_payment_fk FOREIGN KEY (pay_id) REFERENCES
hzb_payment (pay_id);
ALTER TABLE hzb_order
ADD CONSTRAINT hzb_order_hzb_show_fk FOREIGN KEY (show_id) REFERENCES
hzb_show (show_id);
ALTER TABLE hzb_order
ADD CONSTRAINT hzb_order_hzb_store_fk FOREIGN KEY (store_id) REFERENCES

```

```

hzb_store (store_id);
ALTER TABLE hzb_order
ADD CONSTRAINT hzb_order_hzb_tickets_fk FOREIGN KEY (tickets_id)
REFERENCES hzb_tickets (tickets_id);
ALTER TABLE hzb_order
ADD CONSTRAINT hzb_order_hzb_visitor_fk FOREIGN KEY (visitor_id)
REFERENCES hzb_visitor (visitor_id);
ALTER TABLE hzb_parking
ADD CONSTRAINT hzb_parking_hzb_visitor_fk FOREIGN KEY (visitor_id)
REFERENCES hzb_visitor (visitor_id);
ALTER TABLE hzb_tickets
ADD CONSTRAINT hzb_tickets_hzb_visitor_fk FOREIGN KEY (visitor_id)
REFERENCES hzb_visitor (visitor_id);
ALTER TABLE hzb_visit
ADD CONSTRAINT hzb_visit_hzb_attract_fk FOREIGN KEY (attract_id)
REFERENCES hzb_attract (attract_id);
ALTER TABLE hzb_visit
ADD CONSTRAINT hzb_visit_hzb_visitor_fk FOREIGN KEY (visitor_id)
REFERENCES hzb_visitor (visitor_id);
ALTER TABLE hzb_individual
ADD CONSTRAINT individual_hzb_visitor_fk FOREIGN KEY (visitor_id)
REFERENCES hzb_visitor (visitor_id);
ALTER TABLE hzb_member
ADD CONSTRAINT member_hzb_visitor_fk FOREIGN KEY (visitor_id) REFERENCES
hzb_visitor (visitor_id);
ALTER TABLE hzb_school
ADD CONSTRAINT school_hzb_visitor_fk FOREIGN KEY (visitor_id) REFERENCES
hzb_visitor (visitor_id);
DELIMITER / / CREATE TRIGGER arc_fkarc_8_hzb_individual BEFORE
INSERT ON hzb_individual FOR EACH ROW BEGIN
DECLARE d VARCHAR(41);
SELECT a.type INTO d
FROM hzb_visitor a
WHERE a.visitor_id = NEW.visitor_id;
IF (
    d IS NULL
    OR d <> 'Individual'
) THEN SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'FK INDIVIDUAL_HZB_VISITOR_FK in Table HZB_INDIVIDUAL
violates Arc constraint on Table HZB_VISITOR - discriminator column TYPE
doesn't have value 'Individual''';
END IF;
END / / CREATE TRIGGER arc_fkarc_8_hzb_group BEFORE
INSERT ON hzb_group FOR EACH ROW BEGIN
DECLARE d VARCHAR(41);
SELECT a.type INTO d

```

```

FROM hzb_visitor a
WHERE a.visitor_id = NEW.visitor_id;
IF (
    d IS NULL
    OR d <> 'Group'
) THEN SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'FK GROUP_HZB_VISITOR_FK in Table HZB_GROUP violates
Arc constraint on Table HZB_VISITOR - discriminator column TYPE doesn't
have value ''Group''';
END IF;
END //
CREATE TRIGGER arc_fkarc_8_hzb_member BEFORE
INSERT ON hzb_member FOR EACH ROW BEGIN
DECLARE d VARCHAR(41);
SELECT a.type INTO d
FROM hzb_visitor a
WHERE a.visitor_id = NEW.visitor_id;
IF (
    d IS NULL
    OR d <> 'Member'
) THEN SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'FK MEMBER_HZB_VISITOR_FK in Table HZB_MEMBER
violates Arc constraint on Table HZB_VISITOR - discriminator column TYPE
doesn't have value ''Member''';
END IF;
END //
CREATE TRIGGER arc_fkarc_8_hzb_school BEFORE
INSERT ON hzb_school FOR EACH ROW BEGIN
DECLARE d VARCHAR(41);
SELECT a.type INTO d
FROM hzb_visitor a
WHERE a.visitor_id = NEW.visitor_id;
IF (
    d IS NULL
    OR d <> 'School'
) THEN SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'FK SCHOOL_HZB_VISITOR_FK in Table HZB_SCHOOL
violates Arc constraint on Table HZB_VISITOR - discriminator column TYPE
doesn't have value ''School''';
END IF;
END //
DELIMITER ;
ALTER TABLE hzb_show
ADD CONSTRAINT CHK_ShowType CHECK (
    type IN (
        'Drama',

```

```

        'Musical',
        'Comedy',
        'Horror',
        'Adventure'
    )
);
ALTER TABLE hzb_show
ADD CONSTRAINT CHK_WheelchairAccessible CHECK (wchair_access IN ('1',
'0'));
ALTER TABLE hzb_attract
ADD CONSTRAINT CHK_AttractStatus CHECK (
    status IN ('Open', 'Closed', 'Under Maintenance')
);
ALTER TABLE hzb_store
ADD CONSTRAINT CHK_StoreCategory CHECK (
    category IN (
        'Food Stall',
        'Ice Cream Parlor',
        'Restaurant',
        'Gift Shop',
        'Apparels'
    )
);
ALTER TABLE hzb_tickets
ADD CONSTRAINT CHK_TicketMethod CHECK (method IN ('Online', 'Onsite'));
DELIMITER //
CREATE TRIGGER insert_member
AFTER
INSERT ON hzb_visitor FOR EACH ROW BEGIN
DECLARE v_visitor_id INT;
DECLARE v_visit_date DATE;
SET v_visitor_id = NULL;
SET v_visit_date = NULL;
IF NEW.type = 'Member' THEN
SET v_visitor_id = NEW.visitor_id;
SET v_visit_date = NEW.visit_date;
END IF;
IF v_visitor_id IS NOT NULL THEN
INSERT INTO hzb_member (visitor_id, mem_id, mem_date, num)
VALUES (v_visitor_id, v_visitor_id, v_visit_date, 5);
END IF;
END //
DELIMITER ;
DELIMITER //
CREATE TRIGGER insert_group
AFTER

```



```

INSERT ON hzb_visitor FOR EACH ROW BEGIN
DECLARE v_visitor_id INT;
DECLARE v_size INT;
SET v_visitor_id = NULL;
SET v_size = 0;
IF NEW.type = 'Group' THEN
SET v_visitor_id = NEW.visitor_id;
SET v_size = NEW.group_size;
END IF;
IF v_visitor_id IS NOT NULL THEN
INSERT INTO hzb_group (visitor_id, group_size)
VALUES (v_visitor_id, v_size);
END IF;
END //
DELIMITER ;
DELIMITER //
CREATE TRIGGER insert_card
AFTER
INSERT ON hzb_payment FOR EACH ROW BEGIN
IF NEW.pay_method = 'Credit' OR NEW.pay_method = 'Debit' THEN
INSERT INTO hzb_card (pay_id, ch_fname, ch_lname, c_number, exp_date,
cvv, type)
VALUES (NEW.pay_id, NEW.ch_fname, NEW.ch_lname, NEW.c_number,
NEW.exp_date, NEW.cvv, NEW.pay_method);
END IF;
END //
DELIMITER ;
DELIMITER //
CREATE TRIGGER insert_school
AFTER
INSERT ON hzb_visitor FOR EACH ROW BEGIN
DECLARE v_visitor_id INT;
DECLARE v_name VARCHAR(100);
SET v_visitor_id = NULL;
SET v_name = NULL;
IF NEW.type = 'School' THEN
SET v_visitor_id = NEW.visitor_id;
SET v_name = NEW.school_name;
END IF;
IF v_visitor_id IS NOT NULL THEN
INSERT INTO hzb_school (visitor_id, school_id, school_name)
VALUES (v_visitor_id, v_visitor_id, v_name);
END IF;
END //
DELIMITER ;
DELIMITER //

```

```

CREATE TRIGGER insert_individual
AFTER
INSERT ON hzb_visitor FOR EACH ROW BEGIN
DECLARE v_visitor_id INT;
DECLARE v_gender VARCHAR(10);
DECLARE v_nation VARCHAR(15);
IF NEW.type = 'Individual' THEN
SET v_visitor_id = NEW.visitor_id;
SET v_gender = NEW.gender;
SET v_nation = NEW.nationality;
END IF;
IF v_visitor_id IS NOT NULL THEN
INSERT INTO hzb_individual (visitor_id, gender, nationality)
VALUES (v_visitor_id, v_gender, v_nation);
END IF;
END //
DELIMITER ;

DELIMITER //
CREATE TRIGGER apply_discount_on_parking_purchase
AFTER INSERT ON hzb_parking FOR EACH ROW BEGIN
DECLARE v_discount DECIMAL(4, 2) DEFAULT 0;
DECLARE v_discount_num INT DEFAULT 5;
DECLARE v_id INT DEFAULT 0;
DECLARE v_age INT DEFAULT 0;
INSERT INTO hzb_order (
    visitor_id,
    park_id,
    ord_amount,
    ord_date,
    ord_quantity
)
VALUES (
    NEW.visitor_id,
    NEW.park_id,
    NEW.fee * (1 - v_discount),
    NEW.time_in,
    2
);
END //
DELIMITER ;
DELIMITER //
CREATE TRIGGER apply_discount_on_show_purchase BEFORE
INSERT ON hzb_order FOR EACH ROW BEGIN
DECLARE v_discount DECIMAL(4, 2) DEFAULT 0;
DECLARE v_discount_num INT DEFAULT 5;

```

```

DECLARE v_id INT DEFAULT 0;
DECLARE v_age INT DEFAULT 0;
DECLARE s_price DECIMAL(8, 2) DEFAULT 0;
DECLARE sh_time DATE;
SELECT TIMESTAMPDIFF(YEAR, dob, NOW()) INTO v_age
FROM hzb_visitor
WHERE visitor_id = NEW.visitor_id;
IF NEW.show_id IS NOT NULL THEN
SELECT price INTO s_price
FROM hzb_show
WHERE show_id = NEW.show_id;
SELECT s_time INTO sh_time
FROM hzb_show
WHERE show_id = NEW.show_id;
IF v_age < 7 THEN
SET v_discount = 1;
END IF;
SET NEW.ord_amount = s_price * (1 - v_discount) * NEW.ord_quantity;
SET NEW.ord_date = sh_time;
END IF;
END //
DELIMITER ;
DELIMITER //
CREATE TRIGGER apply_discount_on_tickets_purchase
AFTER
INSERT ON hzb_tickets FOR EACH ROW BEGIN
DECLARE v_discount DECIMAL(5, 2) DEFAULT 0;
DECLARE v_discount_num INT DEFAULT 0;
DECLARE v_id INT DEFAULT 0;
DECLARE v_age INT DEFAULT 0;
DECLARE v_quan INT DEFAULT 1;
DECLARE v_type VARCHAR(10) DEFAULT NULL;
DECLARE v_size INT DEFAULT 0;
DECLARE v_amount DECIMAL(8, 2) DEFAULT 0;
IF NEW.method = 'Online' THEN
SET v_discount = 0.05;
END IF;
SELECT type INTO v_type
FROM hzb_visitor
WHERE visitor_id = NEW.visitor_id;
IF v_type = 'Member' THEN
SELECT num INTO v_discount_num
FROM hzb_member
WHERE visitor_id = NEW.visitor_id;
END IF;
IF v_discount_num != 0

```

```

AND v_discount != 0.15 THEN
SET v_discount = v_discount + 0.10;
UPDATE hzb_member
SET num = v_discount_num - 1
WHERE visitor_id = NEW.visitor_id;
END IF;
SELECT TIMESTAMPDIFF(YEAR, dob, NOW()) INTO v_age
FROM hzb_visitor
WHERE visitor_id = NEW.visitor_id;
IF v_age > 60
OR v_age < 7 THEN
SET v_discount = 0.15;
END IF;
IF DAYNAME(NEW.vist_date) IN ('Saturday', 'Sunday')
OR DATE_FORMAT(NEW.vist_date, '%m/%d') IN ('01/01', '06/01', '12/25')
THEN
SET v_discount = 0;
END IF;
SET v_amount = NEW.price * (1 - v_discount);
IF v_type = 'Group' THEN
SELECT group_size INTO v_size
FROM hzb_group
WHERE visitor_id = NEW.visitor_id;
IF NEW.method = 'Online' THEN
SET v_amount = NEW.price * (1 - 0.05) * v_size;
ELSE
SET v_amount = NEW.price * v_size;
END IF;
END IF;
IF v_size = 0 THEN
SET v_size = 1;
END IF;
INSERT INTO hzb_order (
    ord_date,
    ord_quantity,
    ord_amount,
    show_id,
    store_id,
    visitor_id,
    park_id,
    tickets_id
)
VALUES (
    NEW.vist_date,
    v_size,
    v_amount,

```

```

        null,
        null,
        NEW.visitor_id,
        null,
        NEW.tickets_id
    );
END //
DELIMITER ;

DELIMITER //
CREATE TRIGGER apply_discount_on_store_purchase BEFORE
INSERT ON hzb_order FOR EACH ROW BEGIN
DECLARE u_price INT DEFAULT 0;
IF NEW.store_id IS NOT NULL THEN
SELECT unit_price INTO u_price
FROM hzb_store
WHERE store_id = NEW.store_id;
SET NEW.ord_amount := u_price * NEW.ord_quantity;
END IF;
END //
DELIMITER ;

DELIMITER //
CREATE PROCEDURE fetch_data(IN input_date DATE) BEGIN TRUNCATE TABLE
hzb_summ;
INSERT INTO hzb_summ (amount, vist_date, source, source_id, visitor_id)
SELECT ord_amount,
ord_date,
CASE
WHEN show_id IS NOT NULL THEN 'Show'
WHEN store_id IS NOT NULL THEN 'Store'
WHEN park_id IS NOT NULL THEN 'Parking'
WHEN tickets_id IS NOT NULL THEN 'Tickets'
ELSE NULL
END as source,
CASE
WHEN show_id IS NOT NULL THEN show_id
WHEN store_id IS NOT NULL THEN store_id
WHEN park_id IS NOT NULL THEN park_id
WHEN tickets_id IS NOT NULL THEN tickets_id
ELSE NULL
END as source_id,
visitor_id
FROM hzb_order
WHERE ord_date = input_date;
SELECT *

```

```
FROM hzb_summ;
END //
DELIMITER ;
```

Development

Q1) Table joins with at least 3 tables in join

A1) Select Query

```
SELECT v.fname, v.lname, o.show_id, s.name
FROM hzb_visitor v
JOIN hzb_order o ON v.visitor_id = o.visitor_id
JOIN hzb_show s ON o.show_id = s.show_id;
```

A2) Result of the Query

	fname	lname	show_id	name
▶	John	Doe	1	Drama Show 2
	Jane	Doe	2	Comedy Show 3
	Alex	Smith	3	Horror Show 4
	Sam	Jones	4	Adventure Show 5
	Mike	Brown	5	Drama Show 6
	Karen	Lee	6	Comedy Show 8
	Mark	Davis	7	Drama Show 9
	Emily	Kim	8	Musical Show 10
	John	Doe	9	Comedy Show 11
	Jane	Doe	10	Drama Show 6
	Alex	Smith	1	Drama Show 2
	Sam	Jones	2	Comedy Show 3
	Mike	Brown	3	Horror Show 4
	Karen	Lee	4	Adventure Show 5

A3) Brief explanation about the business information you intend to retrieve

This query retrieves the first name, last name, show_id, and show name for visitors who made orders for shows. It combines information from the hzb_visitor, hzb_order, and hzb_show tables using JOIN operations. The result shows the visitor's name, the show_id of the show they ordered, and the name of the show they purchased for.

Q2) Multi-row subquery

A1) Select Query

```
SELECT v.fname, v.lname, o.ord_amount, (SELECT AVG(ord_amount) FROM
 hzb_order) AS 'AVG'
FROM hzb_visitor v
JOIN hzb_order o ON v.visitor_id = o.visitor_id
WHERE o.ord_amount > (SELECT AVG(ord_amount) FROM hzb_order);
```

A2) Result of the Query

	fname	lname	ord_amount	AVG
▶	Jane	Doe	30	23.3333
	Alex	Smith	40	23.3333
	Sam	Jones	50	23.3333
	Karen	Lee	30	23.3333
	Mark	Davis	40	23.3333
	Emily	Kim	50	23.3333
	Karen	Lee	30	23.3333
	Mike	Brown	40	23.3333
	Sam	Jones	50	23.3333
	Alex	Smith	30	23.3333
	Jane	Doe	40	23.3333
	John	Doe	50	23.3333

A3) Brief explanation about the business information you intend to retrieve

This query selects the first name, last name, and order amount for visitors whose order amount is greater than the average order amount of all orders in the hzb_order table. It uses

a multi-row subquery to calculate the average order amount and compares it with the individual order quantities.

Q3) Correlated subquery

A1) Select Query

```
SELECT a.name, a.capacity
FROM hzb_attract a
WHERE a.capacity > (
    SELECT AVG(capacity)
    FROM hzb_attract b
    WHERE a.type = b.type
);
```

A2) Result of the Query

	name	capacity
▶	Roller Coaster 1	50
	Water Ride 1	30
	Dark Ride 1	20
	Kid Ride 1	15
	Water Ride 3	28

A3) Brief explanation about the business information you intend to retrieve

This query retrieves the attraction name and capacity for attractions whose capacity is greater than the average capacity of attractions of the same type. It uses a correlated subquery to compare the capacity of each attraction with the average capacity of attractions with the same type.

Q4) SET operator query

A1) Select Query

```
SELECT fname, lname
FROM hzb_visitor
WHERE type = 'Individual'
UNION
SELECT school_name, ''
FROM hzb_visitor
WHERE type = 'School';
```

A2) Result of the Query

	fname	lname
▶	Jane	Doe
	Karen	Lee
	River	
	South	

A3) Brief explanation about the business information you intend to retrieve

This query combines the first name and last name of individual visitors with the school names of visitors of type 'School' using the UNION operator. The query retrieves a unified list of names for individuals and schools without any duplicate entries.

Q5) Query with in line view or WITH clause

A1) Select Query

```
WITH visitor_purchases AS (
```

```

SELECT v.visitor_id, COUNT(*) AS purchase_count
FROM hzb_visitor v
JOIN hzb_order o ON v.visitor_id = o.visitor_id
GROUP BY v.visitor_id
)
SELECT v.fname, v.lname, vp.purchase_count AS 'Count'
FROM hzb_visitor v
JOIN visitor_purchases vp ON v.visitor_id = vp.visitor_id
WHERE vp.purchase_count >= 3;

```

A2) Result of the Query

	fname	lname	Count
▶	John	Doe	4
	Jane	Doe	4
	Alex	Smith	4
	Sam	Jones	5
	Mike	Brown	4
	Karen	Lee	4
	Mark	Davis	3

A3) Brief explanation about the business information you intend to retrieve

This query uses a WITH clause to create an inline view called visitor_purchases that calculates the purchase count for each visitor based on the number of orders they made. It then joins the hzb_visitor table with the visitor_purchases view to retrieve the first name, last name, and purchase count for visitors. The WHERE clause filters the results to include only visitors with a purchase count of 3 or more.

Q6) TOP-N query

A1) Select Query

```

SELECT fname, lname, visit_date
FROM hzb_visitor
ORDER BY visit_date DESC
LIMIT 5;

```

A2) Result of the Query

	fname	lname	visit_date
▶	Emily	Kim	2022-05-06
	Mark	Davis	2022-05-04
	Karen	Lee	2022-05-03
	Mike	Brown	2022-05-02
	Sam	Jones	2022-05-01

A3) Brief explanation about the business information you intend to retrieve

This query retrieves the top 5 visitors based on their most recent visit dates. The ORDER BY clause sorts the visitors in descending order based on their visit dates. The LIMIT clause is then used to restrict the result set to the first 5 rows, effectively retrieving the top 5 visitors.