

Summary/Abstract

The problem is to meet the guest and host on the same page so that guest can track their transactions, and the host will be able to track their transactions. This project emphasizes solving this problem with a database solution so that data can be persistent and the transaction can be reported or analyzed. In this project, twenty-one tables are designed for demonstration purposes and tables are connected so that technically data can be fetched from each other. The first requirement is specified then related tables are designed with their attributes, and after that Entity relationship diagram is created in the conceptual phase. The development phase contains database generation by SQL (structured data query language) statements and test purposes data will be included in each table. In the development phase, there have been developed slides to show SQL statements for “CREATE”, “INSERT” and “SELECT” commands. In the submitted “.zip” file it may be seen SQL files for installation, each SQL file is separated by a related folder name, and the “SQL_Backup_AirBnB” folder contains the “sql-data.sql” file which contains create and insert statements in one file so that this file can be used for installation to a database. In the finalization phase, all tables with their metadata are added and it can be seen in the table below

No	TABLE_NAME	TABLE_ROWS	DATA_LENGTH	INDEX_LENGTH
1	accountpayable	20	16384	0
2	accountreceivable	20	16384	0
3	accountreceivable_has_facilityinvoice	20	16384	32768
4	bank	20	16384	32768
5	carrier	20	16384	32768
6	employee	20	16384	16384
7	expectedincome	20	16384	49152
8	facility	20	16384	0
9	facilityadvantages	20	16384	16384
10	facilityarchitecture	20	16384	16384
11	facilityavailability	20	16384	16384
12	facilityinvoice	20	16384	49152
13	facilityinvoice_has_accountpayable	20	16384	32768
14	facilitylocation	20	16384	16384
15	facilityorder	20	16384	32768
16	facilityoverdaystayed	20	16384	16384
17	facilityprice	20	16384	16384
18	facilityrate	20	16384	32768
19	facilityreview	20	16384	16384
20	facilightsocial	20	16384	16384
21	facilitytaxrate	20	16384	16384
22	userrole	20	16384	0
23	usertaxid	20	16384	16384

To get meta-data SQL query is used shown below,

SELECT

```
information_schema.TABLES.TABLE_SCHEMA,  
information_schema.TABLES.TABLE_NAME,  
information_schema.TABLES.TABLE_ROWS,  
information_schema.TABLES.DATA_LENGTH,  
information_schema.TABLES.INDEX_LENGTH,  
information_schema.TABLES.TABLE_COMMENT
```

FROM

```
information_schema.TABLES
```

WHERE

```
information_schema.TABLES.TABLE_SCHEMA = 'airbnb' ;
```

There are twenty-three tables shown in the above table but two tables are coming from many-to-many relationships so they are not included in the counting tables in the project. This project is the backbone of the backend in an application which makes the possibility of data persistency and it has the potential to develop a user interface but developing the front-end will not be emphasized. This project helped to gain practical knowledge and gain experience to design databases, specify requirements, and knowledge for CRUD (Create, Read, Update, Delete) applications in the software industry. By doing this project, search engines on the internet are used a lot to dive deep into related topics that are required to build a database.

CONCEPTION PHASE

The project aims to create a database for renting a place to stay like apartments or bedrooms. This database is the data persistence part of the platform users' input, which is used for the back-end. The platform aims to contract between host and guest booking; therefore, the database is based on the user role to keep transactions. Some restrictions need to have functioned with the front-end of the platform, for-example host is not paid until twenty-four hours after the guest is satisfied with what is described on a platform and what s/he sees on the facility. In another example of front-end restrictions functionality, the host has to provide at least one picture and phone, but the guest has to provide more information than that to be a verified user. This project will not focus on how to build front-end applications, but on how to make data persistent by using SQL (Structured data Query Language) based database for a platform that is part of the back-end. The problem description for the project can be summarized as, keeping records of the users. The host has the facility, for example, s/he wants to keep records of the facility locations, guest records, and their details such as invoice order. Also, guest can keep their order records. The advantages of this method is to have historical data of the facility's and historical data of guests'. Disadvantages of this method is that user need to spend some time to enter input and also facility administrator need to spend some time to enter input to the platform as well as this can be employed an extra employee for the facility. After the problem is defined well, there will be ERM (Entity Relationship Model) for the emphasized database. The figure will be placed on the last page of the conception phase. There will be twenty-one entities in the database model, each database has its structured data to increase efficiency so that the use case can be more scalability and can be run faster while querying through SQL. No-SQL based database systems will not be used on this project as SQL based solutions widely used on industry. The list of entities and a short description of tables are shown below:

No	Tables	Description
1	userRole	It defines user role e.g., Guest/Host
2	userTaxId	User Tax Id for invoices
3	bank	To keep record of bank accounts
4	carrier	To keep details of shipment
5	expectedIncome	Host may want to analyze expected and real income
6	facilityRate	To keep record of guest rate for the facility
7	facilityOrder	Facility orders
8	facilityLocation	Geo-location of facility
9	facilitySocial	Saves facility social media links
10	facilityArchitecture	Architectural properties
11	facilityPrice	Keep records of facility price per day
12	facilityInvoice	facility invoice belongs to user
13	facilityAvailability	keep availability dates
14	facility	place to be stay
15	facilityReview	Comments of guest to be saved
16	facilityTaxRate	tax rate for invoice
17	facilityOverDayStayed	guest can stay more than planned as days
18	facilityAdvantages	advantages of facility

19	accountPayable	payments details
20	accountReceivable	receive details
21	employee	employee details

Table 1 List of Entries

These tables are connected so that SQL queries can run via joining tables to each other. Therefore, the conception project is to build the database for CRUD (Create, Read, Update, Delete) operations. However, only Create and Read statements will be used in this project for simplicity and not to complicate the project. After the ERM (Entity Relationship Model) has been built, tables are also identified and designed with their column and data types. All primary keys are unique, not null, and auto increment in each table. Primary keys are mostly used to connect tables so that each row can be identified when SQL queries are running, there will be also foreign keys on the corresponding table while foreign keys are the primary keys of a connected table. MySQL database is used for the database, with MySQL workbench GUI (graphical user interface) for database modeling. The reason of choosing MySQL workbench GUI is to holistically have solution for the project. Int, varchar, timestamp, data types will be used for a variety of input. Relationships are mostly one-to-many but there are also many-to-many relationships that exist. On many-to-many relationships, has an extra table between two tables that will keep records of both primary keys to match records. However, these extra tables from many-to-many relationships are not included in the table list above. As Each table and its columns can be seen below, SQL statements;

userRole		carrier	
id_userRole	INT	id_carrier	INT
userRole	VARCHAR(45)	name	VARCHAR(50)
name	VARCHAR(45)	itemSent	VARCHAR(45)
surname	VARCHAR(45)	trackingNo	VARCHAR(25)
email	VARCHAR(45)	create_time	TIMESTAMP
telephone	VARCHAR(45)	update_time	TIMESTAMP
create_time	TIMESTAMP		
update_time	TIMESTAMP	bank	
		id_bank	INT
userTaxId		bankName	VARCHAR(250)
id_userTaxId	INT	iban	VARCHAR(50)
userTaxNo	VARCHAR(25)	accountHolderNameSurname	VARCHAR(50)
create_time	TIMESTAMP	create_time	TIMESTAMP
update_time	TIMESTAMP	update_time	TIMESTAMP
facilityRate		expectedIncome	
id_facilityRate	INT	id_expectedIncome	INT
rate	INT	expectedIncome	INT
create_time	TIMESTAMP	create_time	TIMESTAMP
update_time	TIMESTAMP	update_time	TIMESTAMP
facilityOrder		facilityLocation	
id_facilityOrder	INT	id_facilityLocation	INT
dayOrder	INT	latitude	VARCHAR(45)
id_userRole	INT	longitude	VARCHAR(45)

create_time	TIMESTAMP	cityName	VARCHAR(45)
update_time	TIMESTAMP	countryName	VARCHAR(45)
		create_time	TIMESTAMP
		update_time	TIMESTAMP
facilitySocial		facilityArchitecture	
id_facilitySocial	INT	id_facilityArchitecture	INT
facebookLink	VARCHAR(150)	hasRoom	INT
instagramLink	VARCHAR(150)	totalSq	INT
create_time	TIMESTAMP	hasBalcoon	TINYINT
update_time	TIMESTAMP	create_time	TIMESTAMP
		update_time	TIMESTAMP
facilityPrice		facilityReview	
id_facilityPrice	INT	id_facilityReview	INT
pricePerDay	INT	guestComment	VARCHAR(150)
create_time	TIMESTAMP	create_time	TIMESTAMP
update_time	TIMESTAMP	update_time	TIMESTAMP
facilityInvoice		accountReceivable	
id_facilityInvoice	INT	id_acctReceivable	INT
stayedDay	INT	amount	INT
dailyPrice	INT	orgName	VARCHAR(50)
create_time	TIMESTAMP	address	VARCHAR(50)
update_time	TIMESTAMP	create_time	TIMESTAMP
		update_time	TIMESTAMP
facility			
id_facility	INT	facilityTaxRate	
type	VARCHAR(45)	id_facilityTaxRate	INT
name	VARCHAR(45)	taxRate	INT
adress	VARCHAR(45)	create_time	TIMESTAMP
email	VARCHAR(45)	update_time	TIMESTAMP
telephone	VARCHAR(45)		
create_time	TIMESTAMP	facilityOverDayStayed	
update_time	TIMESTAMP	id_facilityOverDayStayed	INT
		overDays	INT
employee		create_time	TIMESTAMP
id_employee	INT	update_time	TIMESTAMP
name	VARCHAR(50)		
surname	VARCHAR(50)	facilityAdvantages	
telephone	VARCHAR(50)	id_facilityAdvantage	INT
create_time	TIMESTAMP	advantage	VARCHAR(45)
update_time	TIMESTAMP	create_time	TIMESTAMP
		update_time	TIMESTAMP
accountPayable			
id_acctPayable	INT	facilityAvailability	
amount	INT	id_facilityAvailability	INT
orgName	VARCHAR(50)	availableStartDate	TIMESTAMP
address	VARCHAR(50)	availableEndDate	TIMESTAMP
create_time	TIMESTAMP	create_time	TIMESTAMP
update_time	TIMESTAMP	update_time	TIMESTAMP

Table 2 list of attributes

At the below page it may be seen the Entity Relation Model.

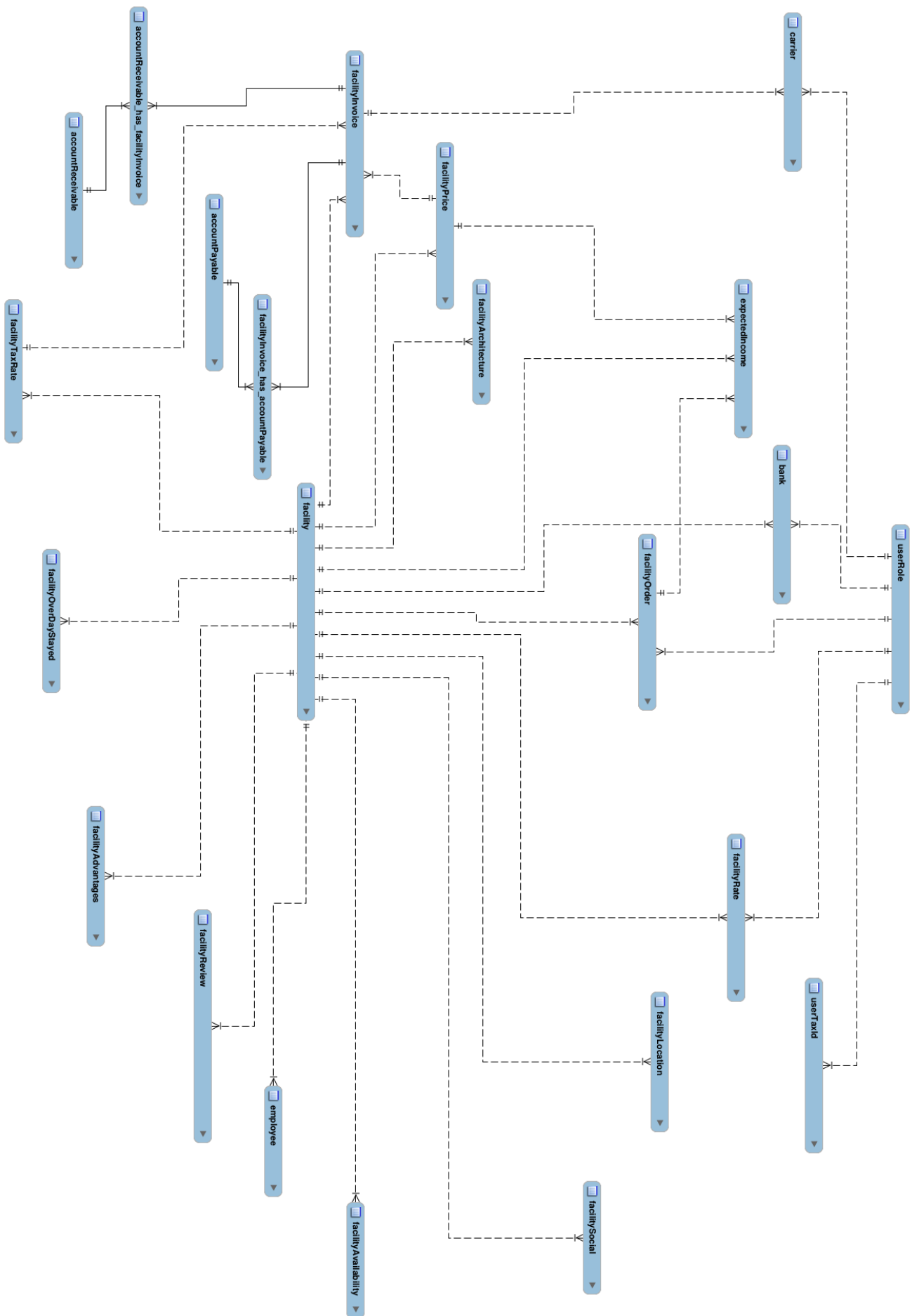


Figure 1 ERM of database

DLBDSPBDM01-Project build a data mart in SQL

use case: AirBnB

Name:	Huseyin
Lastname:	Ozdogan
Marticulation No:	92111877

Implementation procedure: accountPayable table 1/21

Table creation statement :

```
CREATE TABLE `accountPayable` (  
  `id_acctPayable` int NOT NULL AUTO_INCREMENT,  
  `amount` int DEFAULT NULL,  
  `orgName` varchar(50) DEFAULT NULL,  
  `address` varchar(50) DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  PRIMARY KEY (`id_acctPayable`)  
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3  
COMMENT='To define user as Guest or Host';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`accountPayable`  
(`id_acctPayable`,`amount`,`orgName`,`address`,`create_time`,`update_time`)  
VALUES (1,8785922,'Treutel Inc','61950 Wilson Parkway Apt. 499\nSouth Sibyl,  
OH 0958','1995-01-05 12:55:37','1987-10-27 10:59:42');
```

* Data are filled to database are dummy data

Implementation procedure: accountReceivable table 2/21

Table creation statement :

```
CREATE TABLE `accountReceivable` (  
  `id_acctReceivable` int NOT NULL AUTO_INCREMENT,  
  `amount` int DEFAULT NULL,  
  `orgName` varchar(50) DEFAULT NULL,  
  `address` varchar(50) DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  PRIMARY KEY (`id_acctReceivable`)  
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3  
COMMENT='to keep account receivable';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`accountReceivable`  
(`id_acctReceivable`,`amount`,`orgName`,`address`,`create_time`,`update_time`)  
VALUES (1,9211524,'Spinka Inc','3602 Sandra Prairie\nSouth Rosendo, NE  
05968-8299','1986-12-13 07:00:17','1972-06-30 08:21:21');
```

* Data are filled to database are dummy data

Implementation procedure: bank table 3/21

Table creation statement :

```
CREATE TABLE `bank` (  
  `id_bank` int NOT NULL AUTO_INCREMENT,  
  `bankName` varchar(250) DEFAULT NULL,  
  `iban` varchar(50) DEFAULT NULL,  
  `accountHolderNameSurname` varchar(50) DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  `fk_facilityId` int DEFAULT NULL,  
  `fk_userRole` int DEFAULT NULL,  
  PRIMARY KEY (`id_bank`),  
  KEY `fk_bank_1_idx` (`fk_facilityId`),  
  KEY `fk_bank_2_idx` (`fk_userRole`),  
  CONSTRAINT `fk_bank_1` FOREIGN KEY (`fk_facilityId`) REFERENCES `facility` (`id_facility`),  
  CONSTRAINT `fk_bank_2` FOREIGN KEY (`fk_userRole`) REFERENCES `userRole` (`id_userRole`)  
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3 COMMENT='to keep bank records';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`bank`  
(`id_bank`,`bankName`,`iban`,`accountHolderNameSurname`,`create_time`,`update_time`,`fk_facilityId`,`fk_userRole`) VALUES (1,'Smith Group','232710582809','Prof. Jarod Adams III','1994-07-03 01:18:14','1981-08-27 17:17:20',1,1);
```

* Data are filled to database are dummy data

Implementation procedure: carrier table 4/21

Table creation statement :

```
CREATE TABLE `carrier` (  
  `id_carrier` int NOT NULL AUTO_INCREMENT,  
  `name` varchar(50) DEFAULT NULL,  
  `itemSent` varchar(45) DEFAULT NULL,  
  `trackingNo` varchar(25) DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  `fk_facilityInvoiceId` int DEFAULT NULL,  
  `fk_userRoleId` int DEFAULT NULL,  
  PRIMARY KEY (`id_carrier`),  
  KEY `fk_carrier_2_idx` (`fk_facilityInvoiceId`),  
  KEY `fk_carrier_3_idx` (`fk_userRoleId`),  
  CONSTRAINT `fk_carrier_2` FOREIGN KEY (`fk_facilityInvoiceId`) REFERENCES `facilityInvoice`  
  (`id_facilityInvoice`),  
  CONSTRAINT `fk_carrier_3` FOREIGN KEY (`fk_userRoleId`) REFERENCES `userRole` (`id_userRole`)  
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3 COMMENT='to keep records of  
carrier that platform user can track the sent items like invoice';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`carrier`  
(`id_carrier`,`name`,`itemSent`,`trackingNo`,`create_time`,`update_time`,`fk_facilityI  
nvoiceId`,`fk_userRoleId`) VALUES (1,'DHL','invitation','1932617','2003-02-26  
07:09:12','1989-06-25 09:00:36',1,1);
```

* Data are filled to database are dummy data

Implementation procedure: employee table 5/21

Table creation statement :

```
CREATE TABLE `employee` (  
  `id_employee` int NOT NULL AUTO_INCREMENT,  
  `name` varchar(50) DEFAULT NULL,  
  `surname` varchar(50) DEFAULT NULL,  
  `telephone` varchar(50) DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  `fk_facilityId` int DEFAULT NULL,  
  PRIMARY KEY (`id_employee`),  
  KEY `fk_employee_1_idx` (`fk_facilityId`),  
  CONSTRAINT `fk_employee_1` FOREIGN KEY (`fk_facilityId`) REFERENCES `facility`  
  (`id_facility`)  
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3 COMMENT='To define  
user as Guest or Host';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`employee`  
(`id_employee`,`name`,`surname`,`telephone`,`create_time`,`update_time`,`fk_facili  
tyId`) VALUES (1,'Peggie','Bechtelar','1-019-451-9861x0725','2012-05-10  
00:31:06','1986-06-03 01:30:25',1);
```

* Data are filled to database are dummy data

Implementation procedure: expectedIncome table 6/21

Table creation statement :

```
CREATE TABLE `expectedIncome` (  
  `id_expectedIncome` int NOT NULL AUTO_INCREMENT,  
  `expectedIncome` int DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  `fk_facilityId` int DEFAULT NULL,  
  `fk_facilityPricId` int DEFAULT NULL,  
  `fk_facilityOrderId` int DEFAULT NULL,  
  PRIMARY KEY (`id_expectedIncome`),  
  KEY `fk_expectedIncome_1_idx` (`fk_facilityId`),  
  KEY `fk_expectedIncome_2_idx` (`fk_facilityPricId`),  
  KEY `fk_expectedIncome_3_idx` (`fk_facilityOrderId`),  
  CONSTRAINT `fk_expectedIncome_1` FOREIGN KEY (`fk_facilityId`) REFERENCES `facility` (`id_facility`),  
  CONSTRAINT `fk_expectedIncome_2` FOREIGN KEY (`fk_facilityPricId`) REFERENCES `facilityPrice`  
  (`id_facilityPrice`),  
  CONSTRAINT `fk_expectedIncome_3` FOREIGN KEY (`fk_facilityOrderId`) REFERENCES `facilityOrder`  
  (`id_facilityOrder`)  
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3 COMMENT='to keep record for the expected  
income';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`expectedIncome`  
(`id_expectedIncome`,`expectedIncome`,`create_time`,`update_time`,`fk_facilityId`,  
`fk_facilityPricId`,`fk_facilityOrderId`) VALUES (1,4558,'1986-05-14  
16:31:53','1979-05-22 08:51:45',1,1,1);
```

* Data are filled to database are dummy data

Implementation procedure: facility table 7/21

Table creation statement :

```
CREATE TABLE `facility` (  
  `id_facility` int NOT NULL AUTO_INCREMENT,  
  `type` varchar(45) DEFAULT NULL,  
  `name` varchar(45) DEFAULT NULL,  
  `adress` varchar(45) DEFAULT NULL,  
  `email` varchar(45) DEFAULT NULL,  
  `telephone` varchar(45) DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  PRIMARY KEY (`id_facility`)  
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3  
COMMENT='To define facility type like Hotel, apart and etc..';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`facility`  
(`id_facility`, `type`, `name`, `adress`, `email`, `telephone`, `create_time`, `update_time`  
) VALUES (1, 'apartment', 'sint', 'Apt.  
247', 'hhackett@example.com', '08339197935', '1974-08-31 09:18:10', '1970-09-02  
01:18:50');
```

* Data are filled to database are dummy data

Implementation procedure: facilityAdvantages table 8/21

Table creation statement :

```
CREATE TABLE `facilityAdvantages` (  
  `id_facilityAdvantage` int NOT NULL AUTO_INCREMENT,  
  `advantage` varchar(45) DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  `fk_facilityId` int DEFAULT NULL,  
  PRIMARY KEY (`id_facilityAdvantage`),  
  KEY `fk_facilityAdvantages_1_idx` (`fk_facilityId`),  
  CONSTRAINT `fk_facilityAdvantages_1` FOREIGN KEY (`fk_facilityId`)  
    REFERENCES `facility` (`id_facility`)  
  ) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3  
  COMMENT='to keep records for the advantages of facility';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`facilityAdvantages`  
  (`id_facilityAdvantage`,`advantage`,`create_time`,`update_time`,`fk_facilityId`)  
VALUES (1,'Necessitatibus molestias beatae voluptates qu','1972-11-21  
04:41:28','2021-06-27 16:42:03',1);
```

* Data are filled to database are dummy data

Implementation procedure: facilityArchitecture table 9/21

Table creation statement :

```
CREATE TABLE `facilityArchitecture` (  
  `id_facilityArchitecture` int NOT NULL AUTO_INCREMENT,  
  `hasRoom` int DEFAULT NULL,  
  `totalSq` int DEFAULT NULL,  
  `hasBalcoon` tinyint DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  `fk_facilityId` int DEFAULT NULL,  
  PRIMARY KEY (`id_facilityArchitecture`),  
  KEY `fk_facilityArchitecture_1_idx` (`fk_facilityId`),  
  CONSTRAINT `fk_facilityArchitecture_1` FOREIGN KEY (`fk_facilityId`) REFERENCES  
  `facility` (`id_facility`)  
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3 COMMENT='to  
keep record of architectural details';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`facilityArchitecture`  
(`id_facilityArchitecture`,`hasRoom`,`totalSq`,`hasBalcoon`,`create_time`,`update_time`,`fk_facilityId`) VALUES (1,0,80,1,'2000-08-02 23:46:49','2020-05-25 15:19:00',1);
```

* Data are filled to database are dummy data

Implementation procedure: facilityAvailability table 10/21

Table creation statement :

```
CREATE TABLE `facilityAvailability` (  
  `id_facilityAvailability` int NOT NULL AUTO_INCREMENT,  
  `availableStartDate` timestamp NULL DEFAULT NULL,  
  `availableEndDate` timestamp NULL DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  `fk_facilityId` int DEFAULT NULL,  
  PRIMARY KEY (`id_facilityAvailability`),  
  KEY `fk_facilityAvailability_1_idx` (`fk_facilityId`),  
  CONSTRAINT `fk_facilityAvailability_1` FOREIGN KEY (`fk_facilityId`) REFERENCES  
  `facility` (`id_facility`)  
  ) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3 COMMENT='to  
keep records of facility availability days';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`facilityAvailability`  
(`id_facilityAvailability`,`availableStartDate`,`availableEndDate`,`create_time`,`update_time`,`fk_facilityId`) VALUES (1,'1998-02-22 00:31:23','1986-08-05  
19:51:30','1971-06-17 22:59:55','1996-09-14 07:16:02',1);
```

* Data are filled to database are dummy data

Implementation procedure: facilityInvoice table 11/21

Table creation statement :

```
CREATE TABLE `facilityInvoice` (  
  `id_facilityInvoice` int NOT NULL AUTO_INCREMENT,  
  `stayedDay` int DEFAULT NULL,  
  `fk_facilityPriceId` int DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  `fk_facilityId` int DEFAULT NULL,  
  `discount` int DEFAULT NULL,  
  `fk_facilityTaxRateId` int DEFAULT NULL,  
  `invoiceAmountTotal` int DEFAULT NULL,  
  PRIMARY KEY (`id_facilityInvoice`),  
  KEY `fk_facilityInvoice_1_idx` (`fk_facilityPriceId`),  
  KEY `fk_facilityInvoice_2_idx` (`fk_facilityId`),  
  KEY `fk_facilityInvoice_3_idx` (`fk_facilityTaxRateId`),  
  CONSTRAINT `fk_facilityInvoice_1` FOREIGN KEY (`fk_facilityPriceId`) REFERENCES `facilityPrice` (`id_facilityPrice`) ON DELETE  
  CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `fk_facilityInvoice_2` FOREIGN KEY (`fk_facilityId`) REFERENCES `facility` (`id_facility`),  
  CONSTRAINT `fk_facilityInvoice_3` FOREIGN KEY (`fk_facilityTaxRateId`) REFERENCES `facilityTaxRate` (`id_facilityTaxRate`)  
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3 COMMENT='to keep record of invoice belongs to guest';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`facilityInvoice`  
(`id_facilityInvoice`,`stayedDay`,`fk_facilityPriceId`,`create_time`,`update_time`,`fk_`  
`facilityId`,`discount`,`fk_facilityTaxRateId`,`invoiceAmountTotal`) VALUES  
(1,8,3,'1981-07-12 22:07:08','1976-04-27 16:20:32',3,0,3,8);
```

* Data are filled to database are dummy data

Implementation procedure: facilityLocation table 12/21

Table creation statement :

```
CREATE TABLE `facilityLocation` (  
  `id_facilityLocation` int NOT NULL AUTO_INCREMENT COMMENT 'it will keep record of facility location geo  
  coordinations',  
  `latitude` varchar(45) DEFAULT NULL,  
  `longitude` varchar(45) DEFAULT NULL,  
  `cityName` varchar(45) DEFAULT NULL,  
  `countryName` varchar(45) DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  `fk_facilityId` int DEFAULT NULL,  
  PRIMARY KEY (`id_facilityLocation`),  
  KEY `fk_facilityLocation_1_idx` (`fk_facilityId`),  
  CONSTRAINT `fk_facilityLocation_1` FOREIGN KEY (`fk_facilityId`) REFERENCES `facility` (`id_facility`)  
  ON DELETE CASCADE  
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3 COMMENT='to keep record  
  geolocation of facility';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`facilityLocation`  
  (`id_facilityLocation`,`latitude`,`longitude`,`cityName`,`countryName`,`create_time`,`  
  update_time`,`fk_facilityId`) VALUES (1,'-37.839364','  
  76.102502','Wittingberg','French Southern Territories','1977-09-05 06:11:34','2003-  
  08-04 07:07:01',1);
```

* Data are filled to database are dummy data

Implementation procedure: facilityOrder table 13/21

Table creation statement :

```
CREATE TABLE `facilityOrder` (  
  `id_facilityOrder` int NOT NULL AUTO_INCREMENT,  
  `dayOrder` int DEFAULT NULL,  
  `fk_userRoleId` int DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  `fk_facilityId` int DEFAULT NULL,  
  PRIMARY KEY (`id_facilityOrder`),  
  KEY `fk_facilityOrders_1_idx` (`fk_userRoleId`),  
  KEY `fk_facilityOrder_1_idx` (`fk_facilityId`),  
  CONSTRAINT `fk_facilityOrder_1` FOREIGN KEY (`fk_facilityId`) REFERENCES  
  `facility` (`id_facility`),  
  CONSTRAINT `fk_facilityOrders_1` FOREIGN KEY (`fk_userRoleId`) REFERENCES  
  `userRole` (`id_userRole`) ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3  
COMMENT='to keep record of facility order of guest';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`facilityOrder`  
(`id_facilityOrder`,`dayOrder`,`fk_userRoleId`,`create_time`,`update_time`,`fk_facilit  
yId`) VALUES (1,2,1,'2022-04-25 21:52:01','1985-12-15 17:34:53',1);
```

* Data are filled to database are dummy data

Implementation procedure: facilityOverdayStayed table 14/21

Table creation statement :

```
CREATE TABLE `facilityOverDayStayed` (  
  `id_facilityOverDayStayed` int NOT NULL AUTO_INCREMENT,  
  `overDays` int DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  `fk_facilityId` int DEFAULT NULL,  
  PRIMARY KEY (`id_facilityOverDayStayed`),  
  KEY `fk_facilityOverDayStayed_1_idx` (`fk_facilityId`),  
  CONSTRAINT `fk_facilityOverDayStayed_1` FOREIGN KEY (`fk_facilityId`)  
  REFERENCES `facility` (`id_facility`)  
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3  
COMMENT='visitor can stay longer than planned';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`facilityOverDayStayed`  
(`id_facilityOverDayStayed`,`overDays`,`create_time`,`update_time`,`fk_facilityId`)  
VALUES (1,8,'1998-08-16 21:29:21','2008-11-17 08:46:25',1);
```

* Data are filled to database are dummy data

Implementation procedure: facilityPrice table 15/21

Table creation statement :

```
CREATE TABLE `facilityPrice` (  
  `id_facilityPrice` int NOT NULL AUTO_INCREMENT,  
  `pricePerDay` int DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  `fk_facilityId` int DEFAULT NULL,  
  PRIMARY KEY (`id_facilityPrice`),  
  KEY `fk_facilityPrice_1_idx` (`fk_facilityId`),  
  CONSTRAINT `fk_facilityPrice_1` FOREIGN KEY (`fk_facilityId`) REFERENCES  
  `facility` (`id_facility`)  
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3  
COMMENT='to keep records of facility price/day';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`facilityPrice`  
(`id_facilityPrice`,`pricePerDay`,`create_time`,`update_time`,`fk_facilityId`)  
VALUES (1,5,'1998-05-17 20:00:03','1971-05-19 03:36:45',1);
```

* Data are filled to database are dummy data

Implementation procedure: facilityRate table 16/21

Table creation statement :

```
CREATE TABLE `facilityRate` (  
  `id_facilityRate` int NOT NULL AUTO_INCREMENT,  
  `rate` int DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  `fk_userRole` int DEFAULT NULL,  
  `fk_facilityId` int DEFAULT NULL,  
  PRIMARY KEY (`id_facilityRate`),  
  KEY `fk_facilityRate_2_idx` (`fk_userRole`),  
  KEY `fk_facilityRate_1_idx` (`fk_facilityId`),  
  CONSTRAINT `fk_facilityRate_1` FOREIGN KEY (`fk_facilityId`) REFERENCES  
  `facility` (`id_facility`) ON DELETE CASCADE,  
  CONSTRAINT `fk_facilityRate_2` FOREIGN KEY (`fk_userRole`) REFERENCES  
  `userRole` (`id_userRole`) ON DELETE CASCADE  
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3  
COMMENT='To keep record facility rate that user gives';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`facilityRate`  
(`id_facilityRate`,`rate`,`create_time`,`update_time`,`fk_userRole`,`fk_facilityId`)  
VALUES (1,3,'2008-06-28 09:32:51','1973-01-02 22:15:27',1,1);
```

* Data are filled to database are dummy data

Implementation procedure: facilityReview table 17/21

Table creation statement :

```
CREATE TABLE `facilityReview` (  
  `id_facilityReview` int NOT NULL AUTO_INCREMENT,  
  `guestComment` varchar(150) DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  `fk_facilityId` int DEFAULT NULL,  
  PRIMARY KEY (`id_facilityReview`),  
  KEY `fk_facilityReview_1_idx` (`fk_facilityId`),  
  CONSTRAINT `fk_facilityReview_1` FOREIGN KEY (`fk_facilityId`) REFERENCES  
  `facility` (`id_facility`) ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3  
COMMENT='to keep record for the comments of guest to facility';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`facilityReview`  
(`id_facilityReview`,`guestComment`,`create_time`,`update_time`,`fk_facilityId`)  
VALUES (1,'I was a most extraordinary noise going on within--a constant howling  
and sneezing, and every now and then; such as, \'Sure, I don\'t know,\' he went on,  
,\'2012-10-18 03:15:03\',\'1973-03-14 21:27:37\',1);
```

* Data are filled to database are dummy data

Implementation procedure: facilitySocial table 18/21

Table creation statement :

```
CREATE TABLE `facilitySocial` (  
  `id_facilitySocial` int NOT NULL AUTO_INCREMENT,  
  `facebookLink` varchar(150) DEFAULT NULL,  
  `instagramLink` varchar(150) DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  `fk_facilityId` int DEFAULT NULL,  
  PRIMARY KEY (`id_facilitySocial`),  
  KEY `fk_facilitySocial_1_idx` (`fk_facilityId`),  
  CONSTRAINT `fk_facilitySocial_1` FOREIGN KEY (`fk_facilityId`) REFERENCES  
  `facility` (`id_facility`) ON DELETE CASCADE  
  ) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3  
  COMMENT='To keep record social network details';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`facilitySocial`  
(`id_facilitySocial`,`facebookLink`,`instagramLink`,`create_time`,`update_time`,`fk_f  
acilityId`) VALUES (1,'http://www.hyatt.com/','http://runolfsdottirwunsch.com/','1995-  
03-12 12:02:23','2019-04-08 05:17:53',1);
```

* Data are filled to database are dummy data

Implementation procedure: facilityTaxRate table 19/21

Table creation statement :

```
CREATE TABLE `facilityTaxRate` (  
  `id_facilityTaxRate` int NOT NULL AUTO_INCREMENT,  
  `taxRate` int DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  `fk_facilityId` int DEFAULT NULL,  
  PRIMARY KEY (`id_facilityTaxRate`),  
  KEY `fk_facilityTaxRate_1_idx` (`fk_facilityId`),  
  CONSTRAINT `fk_facilityTaxRate_1` FOREIGN KEY (`fk_facilityId`) REFERENCES  
  `facility` (`id_facility`)  
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3  
COMMENT='each facility can have it own tax rate according to regulations';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`facilityTaxRate`  
(`id_facilityTaxRate`,`taxRate`,`create_time`,`update_time`,`fk_facilityId`) VALUES  
(1,5,'1993-09-14 18:20:57','1978-08-03 16:20:00',1);
```

* Data are filled to database are dummy data

Implementation procedure: userRole table 20/21

Table creation statement :

```
CREATE TABLE `userRole` (  
  `id_userRole` int NOT NULL AUTO_INCREMENT,  
  `userRole` varchar(45) DEFAULT NULL,  
  `name` varchar(45) DEFAULT NULL,  
  `surname` varchar(45) DEFAULT NULL,  
  `email` varchar(45) DEFAULT NULL,  
  `telephone` varchar(45) DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  PRIMARY KEY (`id_userRole`)  
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3  
COMMENT='To define user as Guest or Host';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`userRole`  
(`id_userRole`,`userRole`,`name`,`surname`,`email`,`telephone`,`create_time`,`update_time`) VALUES (1,'host','et','voluptatem','jrolfson@example.org','(606)634-1706x00704','2000-08-25 06:12:13','2004-10-18 04:12:38');
```

* Data are filled to database are dummy data

Implementation procedure: userTaxId table 21/21

Table creation statement :

```
CREATE TABLE `userTaxId` (  
  `id_userTaxId` int NOT NULL AUTO_INCREMENT,  
  `userTaxNo` varchar(25) DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT NULL,  
  `update_time` timestamp NULL DEFAULT NULL,  
  `fk_userRoleId` int DEFAULT NULL,  
  PRIMARY KEY (`id_userTaxId`),  
  KEY `fk_userTaxId_1_idx` (`fk_userRoleId`),  
  CONSTRAINT `fk_userTaxId_1` FOREIGN KEY (`fk_userRoleId`) REFERENCES  
  `userRole` (`id_userRole`)  
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb3  
COMMENT='to keep records of the guest tax records';
```

1st/20 Insert statement :

```
INSERT INTO `AirBnB`.`userTaxId`  
(`id_userTaxId`,`userTaxNo`,`create_time`,`update_time`,`fk_userRoleId`) VALUES  
(1,'90874-8312','2012-01-15 13:43:00','1997-12-31 05:12:11',1);
```

* Data are filled to database are dummy data

Ex. Query1: Amount need to be paid from which guest and guest stayed where?

```
SELECT
    airbnb.facilityinvoice.create_time AS 'Invoice date', airbnb.facilityinvoice.stayedDay AS Stayed_Day, airbnb.facilityprice.pricePerDay AS 'Price/Day',
    airbnb.accountpayable.amount AS Amount, airbnb.facilityinvoice.discount AS Discount, airbnb.facilityinvoice.invoiceAmountTotal AS 'Total Invoice Amount',
    airbnb.facility.name AS Host_Name, airbnb.facility.adress AS Host_Address, airbnb.accountpayable.orgName AS Guest_Name,
    airbnb.accountpayable.address AS Guest_Address
FROM airbnb.facilityinvoice_has_accountpayable
INNER JOIN airbnb.accountpayable ON ( airbnb.facilityinvoice_has_accountpayable.accountPayable_id_acctPayable = airbnb.accountpayable.id_acctPayable)
INNER JOIN airbnb.facilityinvoice ON ( airbnb.facilityinvoice_has_accountpayable.facilityInvoice_id_facilityInvoice = airbnb.facilityinvoice.id_facilityInvoice)
INNER JOIN airbnb.facility ON ( airbnb.facilityinvoice.fk_facilityId = airbnb.facility.id_facility)
INNER JOIN airbnb.facilityprice ON ( airbnb.facilityinvoice.fk_facilityPriceId = airbnb.facilityprice.id_facilityPrice) ;
```

Have joined 5 tables together. Inner join method is used to query tables,

Ex. Query1 (cont.): Result

	A	B	C	D	E	F	G	H	I	J
1	Invoice date	Stayed_Day	Price/Day	Amount	Discount	Total Invoice Amount	Host_Name	Host_Address	Guest_Name	Guest_Address
2	1991-04-05 19:48:34	1	7	8785922	5	9	iure	Suite 392	Treutel Inc	61950 Wilson Parkway Apt. 499South Sibyl, OH 0958
3	1991-03-10 12:27:23	2	2	1839387	5	7	voluptate	Apt. 959	Bailey, Cormier and Wiza	894 Kendall TrailHerzogton, NY 17795-4025
4	2017-03-04 06:05:03	3	3	1839387	7	9	quae	Suite 027	Bailey, Cormier and Wiza	894 Kendall TrailHerzogton, NY 17795-4025
5	1995-09-27 06:33:35	1	9	1145380	9	3	consequatur	Suite 166	VonRueden Ltd	76434 Braun StreetWest Jettmouth, WI 83879
6	2004-03-18 15:48:29	6	2	1145380	0	1	incidunt	Apt. 285	VonRueden Ltd	76434 Braun StreetWest Jettmouth, WI 83879
7	2009-08-07 07:04:45	2	3	8054806	5	6	rem	Apt. 885	Runolfsdottir PLC	52882 Benton KnollEast Lavada, TX 66574
8	1978-10-27 03:35:27	9	2	8054806	7	7	ea	Apt. 595	Runolfsdottir PLC	52882 Benton KnollEast Lavada, TX 66574
9	1981-07-12 22:07:08	8	4	9705992	0	8	ipsum	Apt. 961	Wolf Inc	8946 Kerluke VistaHoracechester, MD 33443
10	1988-05-21 18:28:46	5	7	9705992	7	1	quo	Apt. 193	Wolf Inc	8946 Kerluke VistaHoracechester, MD 33443
11	1996-08-09 04:10:17	5	2	8399297	8	3	perspiciatis	Apt. 724	Adams-Renner	4180 Bode Rue Suite 007West Darleneton, ME 42943-
12	1979-10-19 02:01:40	1	3	6114609	2	9	quae	Apt. 711	Grimes PLC	44391 Koch Drive Apt. 993South Jeffereychester, N
13	2004-08-10 07:16:32	2	3	8230085	8	4	commodi	Suite 857	Robel, Kiehn and Kuhic	354 Beatty Green Apt. 588Heidenreichville, NY 413
14	2011-02-05 00:34:31	6	1	4317943	6	9	et	Apt. 108	Rempel Inc	430 Nicholas Forks Apt. 238Isadoreland, MS 70030-
15	1979-07-19 02:29:45	7	5	6804273	0	6	sint	Apt. 247	Orn-Aufderhar	55335 Connelly CrestEast Irmaborough, CA 71107-05
16	1970-03-25 16:21:28	4	1	8585218	7	6	eum	Apt. 441	Ferry Group	32840 Sonya LaneSouth Macie, NV 22273
17	2015-09-25 10:22:53	1	5	8585218	8	3	debitis	Apt. 651	Ferry Group	32840 Sonya LaneSouth Macie, NV 22273
18	1992-02-13 16:26:28	2	2	7154032	4	9	aut	Apt. 260	Kuhic-Reinger	91568 Dell ManorsAndersonshire, NV 36253-3920
19	2005-03-06 02:29:19	5	4	7303558	5	2	sunt	Apt. 652	McCullough, Harris and Kihn	0501 Arielle ParkwayQuigleychester, VA 65109-7690
20	1992-04-01 02:30:29	3	1	2398974	5	0	commodi	Suite 667	Collins, Bailey and Mills	72467 Lockman ShoreEast Destinee, UT 63238
21	1996-03-18 04:38:44	3	9	7062881	8	1	nam	Suite 293	Dooley-Willms	00275 Camilla StravenueSouth Fredericmouth, ID 99

Ex. Query2: which facility is ordered by whom, when, reservation day, guest bank details?

SELECT

```
airbnb.userrole.userRole          AS 'Role',
CONCAT(airbnb.userrole.name, ' ', airbnb.userrole.surname) AS USER_NAME_SURNAME,
airbnb.facility.name              AS Host_Name,
airbnb.facilityorder.dayOrder     AS 'Ordered/Day',
airbnb.facilityorder.create_time  AS 'Order Date',
airbnb.usertaxid.userTaxNo        AS Tax_ID,
airbnb.bank.bankName             AS Bank_Name,
airbnb.bank.iban                 AS IBAN
```

FROM airbnb.bank

INNER JOIN airbnb.userrole **ON** (airbnb.bank.fk_userRole = airbnb.userrole.id_userRole)

INNER JOIN airbnb.facility **ON** (airbnb.bank.fk_facilityId = airbnb.facility.id_facility)

INNER JOIN airbnb.facilityorder **ON** (airbnb.facility.id_facility = airbnb.facilityorder.fk_facilityId) , airbnb.usertaxid

WHERE airbnb.userrole.userRole = 'guest' ;

Have joined 5 tables together with where condition. Inner join method is used to query tables, CONCAT built-in function is used to merge name and surname columns

Ex. Query2 (cont.): Result

	A	B	C	D	E	F	G	H
1	Role	USER_NAME_SURNAME	Host_Name	Ordered/Day	Order Date	Tax_ID	Bank_Name	IBAN
2	guest	assumenda tempora	perspiciatis	4	1973-05-21 01:16:03	90874-8312	Johns PLC	917610177909
3	guest	enim temporibus	commodi	9	1977-08-16 14:18:36	90874-8312	Swaniawski, Sauer and Grimes	813110936665
4	guest	ipsa delectus	nam	7	1972-06-06 04:27:02	90874-8312	Goodwin-Nikolaus	785970765445
5	guest	quia eaque	quo	3	1999-06-20 14:53:42	90874-8312	Littel-Rosenbaum	333704898552
6	guest	et sed	ea	3	1973-12-25 05:42:52	90874-8312	Koelpin-Kunde	496180344140
7	guest	eos reiciendis	quae	9	1970-04-28 17:45:21	90874-8312	Lindgren Ltd	939681075932
8	guest	sed hic	et	8	2015-10-14 07:02:33	90874-8312	Bradtke-Legros	190790513716
9	guest	optio dolores	rem	6	2009-06-21 20:11:18	90874-8312	Prohaska-Feil	965751768276
10	guest	nemo autem	iure	6	2021-07-26 18:28:11	90874-8312	Wuckert, Huels and Smith	718522131256
11	guest	ipsa sed	commodi	8	1978-11-14 18:37:25	90874-8312	Hodkiewicz Inc	971773613756
12	guest	ipsa mollitia	aut	2	1972-02-19 23:26:42	90874-8312	Turcotte-Lowe	788009497430
13	guest	ipsum voluptatum	ipsum	8	1984-05-04 00:41:52	90874-8312	Kovacek, Mosciski and Mosciski	683272848837
14	guest	illo et	incidunt	3	1992-06-22 21:40:09	90874-8312	Lang PLC	734831648506
15	guest	assumenda tempora	perspiciatis	4	1973-05-21 01:16:03	38742	Johns PLC	917610177909
16	guest	enim temporibus	commodi	9	1977-08-16 14:18:36	38742	Swaniawski, Sauer and Grimes	813110936665
17	guest	ipsa delectus	nam	7	1972-06-06 04:27:02	38742	Goodwin-Nikolaus	785970765445
18	guest	quia eaque	quo	3	1999-06-20 14:53:42	38742	Littel-Rosenbaum	333704898552
19	guest	et sed	ea	3	1973-12-25 05:42:52	38742	Koelpin-Kunde	496180344140
20	guest	eos reiciendis	quae	9	1970-04-28 17:45:21	38742	Lindgren Ltd	939681075932
21	guest	sed hic	et	8	2015-10-14 07:02:33	38742	Bradtke-Legros	190790513716

Ex. Query3: Host available dates and architecture details and Host type

```
SELECT
    airbnb.facility.type AS Type, airbnb.facility.name AS Host_Name, airbnb.facility.adress AS Host_Address,   airbnb.facilityarchitecture.totalSq AS Total_Sq,
    airbnb.facilityarchitecture.hasBalcoon AS Has_Balcoon,   airbnb.facilityarchitecture.hasRoom AS Has_Room,
    airbnb.facilityavailability.availableStartDate AS Availibility_Start_Date,   airbnb.facilityavailability.availableEndDate AS Availibility_End_Date
FROM   airbnb.facilityarchitecture
INNER JOIN   airbnb.facility ON   ( airbnb.facilityarchitecture.fk_facilityId = airbnb.facility.id_facility)
INNER JOIN   airbnb.facilityavailability ON   (      airbnb.facility.id_facility = airbnb.facilityavailability.fk_facilityId) ;
```

Have joined 3 tables together. Inner join method is used to query tables,

Ex. Query3 (cont.): Result

	A	B	C	D	E	F	G	H
1	Type	Host_Name	Host_Address	Total_Sq	Has_Balcoon	Has_Room	Availibility_Start_Date	Availibility_End_Date
2	apartment	sint	Apt. 247	80	1	0	1998-02-22 01:31:23	1986-08-05 19:51:30
3	suite	incidunt	Apt. 285	41	0	0	1979-10-19 15:38:07	1979-07-30 05:37:36
4	apartment	ipsum	Apt. 961	37	1	0	2004-03-31 19:30:10	1984-03-22 10:48:26
5	apartment	quae	Apt. 711	37	0	0	2011-06-20 16:50:53	1980-12-27 04:02:01
6	apartment	aut	Apt. 260	71	1	1	2011-01-24 00:43:03	1975-04-03 09:50:44
7	bedroom	commodi	Suite 667	70	0	0	1990-10-18 00:34:09	1984-02-22 21:27:55
8	apartment	sunt	Apt. 652	39	0	1	1984-10-14 01:50:02	2008-06-07 03:36:54
9	bedroom	iure	Suite 392	80	0	0	2016-11-22 20:03:48	2005-07-14 18:14:45
10	apartment	voluptate	Apt. 959	51	0	0	1996-10-27 07:43:15	1982-10-14 05:56:40
11	bedroom	rem	Apt. 885	40	0	0	2000-07-20 20:40:15	2008-09-14 08:12:32
12	suite	et	Apt. 108	52	0	1	1977-01-09 04:50:08	1977-03-06 13:12:13
13	apartment	eum	Apt. 441	25	0	0	2021-05-10 06:54:10	2017-07-04 06:27:31
14	bedroom	quae	Suite 027	50	0	0	2010-02-09 10:51:00	1997-02-27 03:26:55
15	bedroom	ea	Apt. 595	62	0	0	1991-05-03 03:32:59	1989-05-20 22:06:17
16	suite	quo	Apt. 193	41	1	1	2003-06-25 22:02:20	1970-06-18 02:38:17
17	apartment	nam	Suite 293	75	0	0	1976-07-10 08:32:00	1988-01-02 06:45:45
18	suite	debitis	Apt. 651	66	1	1	1975-04-25 21:46:38	2018-02-01 23:25:45
19	bedroom	commodi	Suite 857	38	1	0	1979-08-20 08:10:49	2017-05-05 14:45:11
20	bedroom	perspiciatis	Apt. 724	78	0	1	1985-12-24 04:22:09	2012-06-16 10:04:32
21	bedroom	consequatur	Suite 166	33	0	1	1970-07-17 04:23:27	2007-04-16 18:07:04

Thank you

To find metadata of tables, “SELECT” statement is used as seen below;

SELECT

```
information_schema.TABLES.TABLE_SCHEMA,  
information_schema.TABLES.TABLE_NAME,  
information_schema.TABLES.TABLE_ROWS,  
information_schema.TABLES.DATA_LENGTH,  
information_schema.TABLES.INDEX_LENGTH,  
information_schema.TABLES.TABLE_COMMENT
```

FROM

```
information_schema.TABLES
```

WHERE

```
information_schema.TABLES.TABLE_SCHEMA = 'airbnb' ;
```

result is;

TABLE_NAME	TABLE_ROWS	DATA_LENGTH	INDEX_LENGTH
accountpayable	20	16384	0
accountreceivable	20	16384	0
accountreceivable_has_facilityinvoice	20	16384	32768
bank	20	16384	32768
carrier	20	16384	32768
employee	20	16384	16384
expectedincome	20	16384	49152
facility	20	16384	0
facilityadvantages	20	16384	16384
facilityarchitecture	20	16384	16384
facilityavailability	20	16384	16384
facilityinvoice	20	16384	49152
facilityinvoice_has_accountpayable	20	16384	32768
facilitylocation	20	16384	16384
facilityorder	20	16384	32768
facilityoverdaystayed	20	16384	16384
facilityprice	20	16384	16384
facilityrate	20	16384	32768
facilityreview	20	16384	16384
facilitysocial	20	16384	16384
facilitytaxrate	20	16384	16384
userrole	20	16384	0
usertaxid	20	16384	16384