0. Pre-Illumination

In this project report we will follow the requirement of Phase IV directly. In Section 1 we gave problem description copied from Web site; in Section 2 we answered 3 questions listed in the project and justified our solution; in Section 3 we exhibited EER diagram with all assumptions; in Section 4 we showed our relational schema after normalization; in Section 5 we gave all requested SQL statements for both views and queries; and in Section 6 we gave dependency diagram induced from relational schemas. Finally, a short summary is given at the end of this report.

1. Problem Description

See the Phase 0.

2. Three Questions

2.1 1. Is the ability to model super-class / subclass relationships likely to be important in such environment? Why or why not?

Ans: Super-class, subclass is very important in our database design. For example, in the hotel, we have lots of different types, such as dining, concierge, housekeeping, etc. Without the super-class/ subclass, we would have to use weak entity type to represent the different types, which could lead to redundance. Now, we use subclasses of employee types and thus, our database achieves BCNF.

2.2

1. There are two rates of membership: Silver Prime & Golden Prime
2. There is another type of employee, shuttle driver. A shuttle driver has an experience year.
3. The hotel has an account in the bank. The accountant either has the authorization to manage this account or not. The account records the profit and cost daily.
4. There is a type of employee as procurement. They purchase goods needed for the hotel.
5. There is a type of employee as advertisement. A member in this apartment has an experience year.

2.3 Justify using a Relational DBMS like Oracle for this project.

See figure 1.

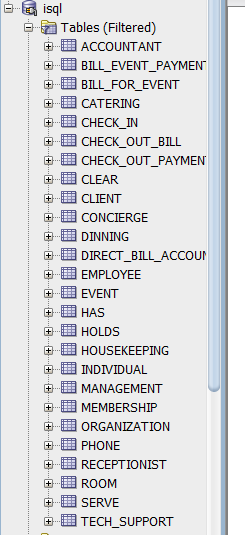
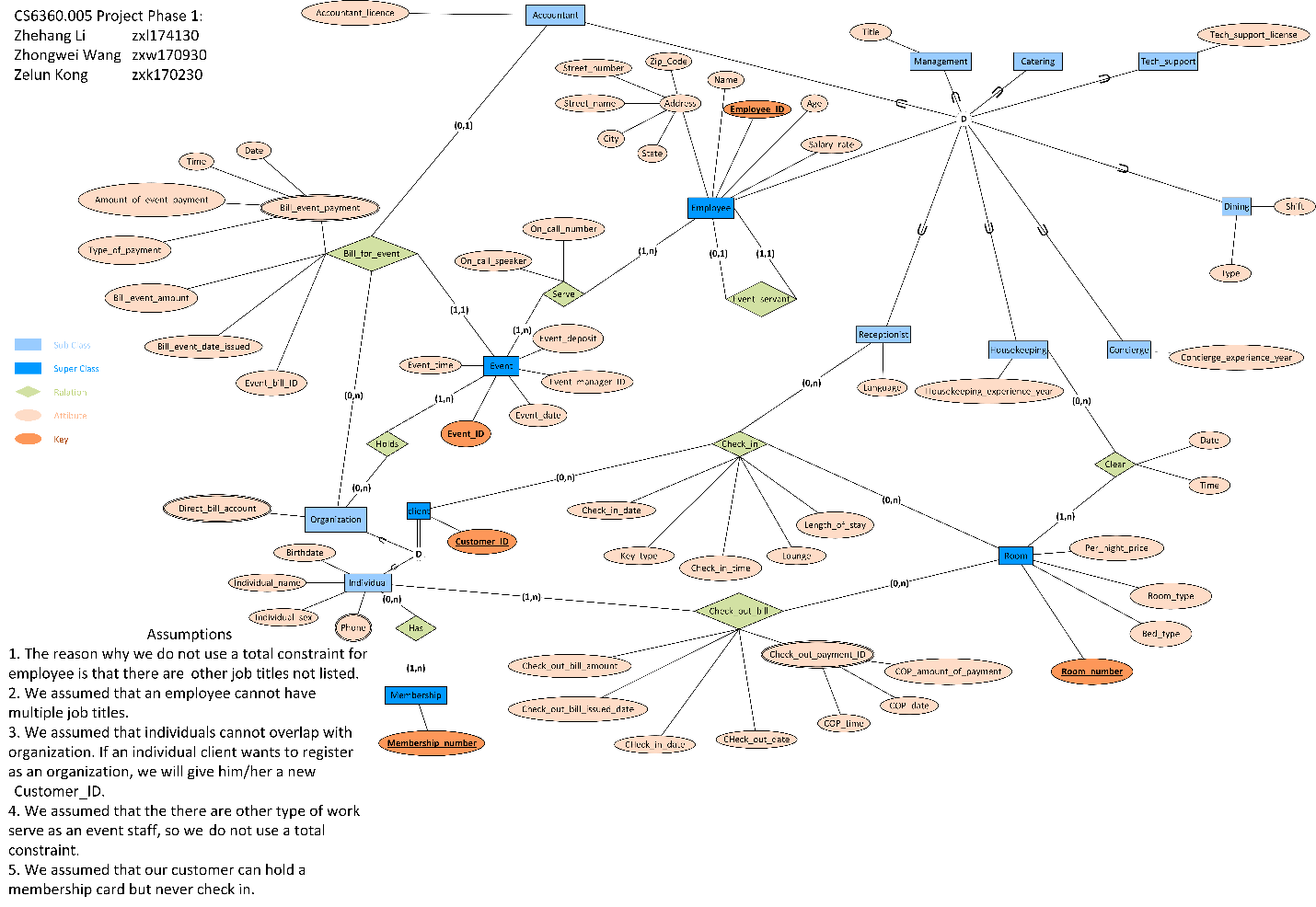


Figure 1: Oracle database implementation for the hotel database design.

3. EER diagram with all assumptions



4. Relational Schema in Third Normal Form

4.1 Regular Entity

1. Employee:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| EmployeeID | Age | Salary\_rate | Name | State | City | Street\_name | Street\_number | Zipcode |

1. Dinning:

|  |  |  |
| --- | --- | --- |
| EmployeeID | Shift | Type |

EmployeeID is a FK referencing to the Employee.EmployeeID.

1. Concierge:

|  |  |
| --- | --- |
| EmployeeID | Concierge\_experience\_year |

EmployeeID is a FK referencing to the Employee.EmployeeID.

1. Housekeeping:

|  |  |
| --- | --- |
| EmployeeID | Housekeeping\_experience\_year |

EmployeeID is a FK referencing to the Employee.EmployeeID.

1. Receptionist:

|  |  |
| --- | --- |
| EmployeeID | Language |

EmployeeID is a FK referencing to the Employee.EmployeeID.

1. Tech\_support:

|  |  |
| --- | --- |
| EmployeeID | Tech\_support\_licence |

EmployeeID is a FK referencing to the Employee.EmployeeID.

1. Management:

|  |  |
| --- | --- |
| EmployeeID | Title |

EmployeeID is a FK referencing to the Employee.EmployeeID.

1. Catering:

|  |
| --- |
| EmployeeID |

EmployeeID is a FK referencing to the Employee.EmployeeID.

1. Accountant:

|  |  |
| --- | --- |
| EmployeeID | Accountant\_licence |

EmployeeID is a FK referencing to the Employee.EmployeeID.

1. Event:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Event\_ID | Event\_date | Event\_time | Event\_manager\_ID | Event\_deposit |

Event\_manager\_ID is a FK referencing to the Management.EmployeeID.

1. Client:

|  |
| --- |
| Customer\_ID |

1. Organization:

|  |
| --- |
| Customer\_ID |

Customer\_ID is a FK referencing to the Client. Customer\_ID.

1. Individual:

|  |  |  |  |
| --- | --- | --- | --- |
| Customer\_ID | Birthdate | Individual\_name | Individual\_sex |

Customer\_ID is a FK referencing to the Client. Customer\_ID.

1. Membership:

|  |
| --- |
| Membership\_number |

1. Room:

|  |  |  |  |
| --- | --- | --- | --- |
| Room\_number | Bed\_type | Room\_type | Per\_night\_price |

2.2: Weak Entity (We do not have this☹)

2.3: Mapping 1:1 relationship

Employee:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| EmployeeID | Age | Salary\_rate | Name | State | City | Street\_name | Street\_number | Zipcode | Event\_staff\_ID |

Event\_staff\_ID is a FK referencing to the Employee. EmployeeID.

2.4 Mapping 1:N relationship (We do not have this☹)

2.5 Mapping M:N relationship

1. Clear:

|  |  |  |  |
| --- | --- | --- | --- |
| Employee\_ID | Room\_number | Date | Time |

Employee\_ID is a FK referencing to the Housekeeping.EmployeeID.

Room\_number is a FK referencing to the Room. Room\_number

1. Check\_out\_bill:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Room\_number | Customer\_ID | Check\_out\_bill\_amount | Check\_out\_bill\_issued\_date | Check\_in\_date | Check\_out\_date |

Room\_number is a FK referencing to the Room. Room\_number.

Customer\_ID is a FK referencing to the Individual. Customer\_ID.

1. Has:

|  |  |
| --- | --- |
| Customer\_ID | Membership\_number |

Customer\_ID is a FK referencing to the Individual. Customer\_ID.

Membership\_number is a FK referencing to the Membership. Membership\_number.

1. Holds:

|  |  |
| --- | --- |
| Event\_ID | Customer\_ID |

Event\_ID is a FK referencing to the Event.Event\_ID.

Customer\_ID is a FK referencing to the Organization.Customer\_ID.

1. Serve:

|  |  |  |  |
| --- | --- | --- | --- |
| Event\_ID | Employee\_ID | On\_call\_speaker | On-call\_number |

Event\_ID is a FK referencing to the Event.Event\_ID.

Employee\_ID is a FK referencing to the Employee.Employee\_ID.

2.6: Mapping of Multivalued Attributes

1. Bill\_event\_payment:

(Implemented after 2.7)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Customer\_ID | Event\_bill\_ID | Date | Time | Amount\_of\_event\_payment | Date\_of\_payment |

Customer\_ID is a FK referencing to the Organization.Customer\_ID.

1. Direct\_bill\_account:

|  |  |
| --- | --- |
| Customer\_ID | Direct\_bill\_account |

Customer\_ID is a FK referencing to the Organization.Customer\_ID.

1. Phone:

|  |  |
| --- | --- |
| Customer\_ID | Phone |

Customer\_ID is a FK referencing to the Individual.Customer\_ID.

1. Check\_out\_payment\_ID:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Customer\_ID | Room\_number | COP\_time | COP\_date | COP\_amount\_of\_payment |

Customer\_ID is a FK referencing to the Individual.Customer\_ID.

Room\_number is a FK referencing to the Room.Room\_number.

2.7: Mapping n-ary relationship

1. Check\_in:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Customer\_ID | Employee\_ID | Room\_number | Check\_in\_date | Key\_type | Check\_in\_time | Lounge | Length\_of\_stay |

Customer\_ID is a FK referencing to the Client.Customer\_ID.

Employee\_ID is a FK referencing to the Receptionist.Employee\_ID.

Room\_number is a FK referencing to the Room.Room\_number.

1. Bill\_for\_event:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Customer\_ID | Event\_bill\_ID | Event\_ID | Employee\_ID | Bill\_event\_date\_issued | Bill\_event\_amount |

Customer\_ID is a FK referencing to the Organization.Customer\_ID.

Event\_ID is a FK referencing to the Event.Event\_ID.

Employee\_ID is a FK referencing to the Accountant.Employee\_ID.

4.1 Explanation for format design

Assumption:

* Data format for all Employee ID is an integer with length of 9.
* Age is an integer without any further restriction.
* Salary\_rate is a number within [0, 9999999.99].
* Event\_deposit is a number within [0, 99999.99]. (Max deposit is roughly one hundred thousand dollars.)
* Room number is a 4 digits number.
* Membership number is a 11 digits number.
* Direct bill account is a 20 char-long string.
* Length of stay is an integer greater than 0.

4.2 Format for Every Relation

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Employee | EmployeeID | Numeric(9,0) |
| Age | Integer |
| Salary\_rate | Numeric(9,2) |
| Name | String<=20 |
| State | String<=2 |
| City | String<=20 |
| Street\_name | String<=20 |
| Street\_number | String<=20 |
| Zipcode | String<=10 |
| Event\_staff\_ID | Numeric(9,0) |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Dinning | EmployeeID | Numeric(9,0) |
| Shift | String<=10 |
| Type | String<=20 |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Concierge | EmployeeID | Numeric(9,0) |
| Concierge\_experience\_year | Interger |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Housekeeping | EmployeeID | Numeric(9,0) |
| Housekeeping\_experience\_year | Interger |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Receptionist | EmployeeID | Numeric(9,0) |
| Language | String<=10 |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Tech\_support | EmployeeID | Numeric(9,0) |
| Tech\_support\_licence | String<=100 |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Management | EmployeeID | Numeric(9,0) |
| Title | String<=20 |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Catering | EmployeeID | Numeric(9,0) |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Accountant | EmployeeID | Numeric(9,0) |
| Accountant\_licence | String<=20 |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Event | Event\_ID | Numeric(4,0) |
| Event\_date | MM/DD/YYYY, STRING = 10 CHARS |
| Event\_time | hh-mm-ss, string = 8 chars |
| Event\_manager\_ID | Numeric(9,0) |
| Event\_deposit | Numeric(7,2) |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Client | CustomerID | Numeric(6,0) |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Organization | CustomerID | Numeric(6,0) |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Individual | CustomerID | Numeric(6,0) |
| Birthdate | MM/DD/YYYY, STRING = 10 CHARS |
| Individual\_name | String<=20 |
| Individual\_sex | String<=20 |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Membership | Membership\_number | Numeric(11,0) |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Room | Room\_number | Numeric(4,0) |
| Bed\_type | String<=20 |
| Room\_type | String<=20 |
| Per\_night\_price | Integer |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Clear | EmployeeID | Numeric(9,0) |
| Room\_number | Numeric(4,0) |
| cDate | MM/DD/YYYY, STRING = 10 CHARS |
| cTime | hh-mm-ss, string = 8 chars |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Check\_out\_bill | Room\_number | Numeric(4,0) |
| CustomerID | Numeric(6,0) |
| Check\_out\_bill\_amount | Integer |
| Check\_out\_bill\_issued\_date | MM/DD/YYYY, STRING = 10 CHARS |
| Check\_in\_date | MM/DD/YYYY, STRING = 10 CHARS |
| Check\_out\_date | MM/DD/YYYY, STRING = 10 CHARS |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Has | CustomerID | Numeric(6,0) |
| Membership\_number | Numeric(11,0) |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Holds | CustomerID | Numeric(6,0) |
| Event\_ID | Numeric(4,0) |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Serve | Event\_ID | Numeric(4,0) |
| EmployeeID | Numeric(9,0) |
| On\_call\_speaker | String<=20 |
| On-call\_number | Numeric(4,0) |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Bill\_event\_payment | CustomerID | Numeric(6,0) |
| Event\_bill\_ID | Numeric(6,0) |
| pDate | MM/DD/YYYY, STRING = 10 CHARS |
| pTime | hh-mm-ss, string = 8 chars |
| Amount\_of\_event\_payment | Numeric(8,2) |
| Date\_of\_payment | MM/DD/YYYY, STRING = 10 CHARS |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Direct\_bill\_account | CustomerID | Numeric(6,0) |
| Direct\_bill\_account | String=20 |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Phone | CustomerID | Numeric(6,0) |
| Phone | Numeric(10,0) |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| check\_out\_payment\_id | CustomerID | Numeric(6,0) |
| Room\_number | Numeric(4,0) |
| COP\_time | hh-mm-ss, string = 8 chars |
| COP\_date | MM/DD/YYYY, STRING = 10 CHARS |
| COP\_amount\_of\_payment | Numeric(8,2) |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Check\_in | CustomerID | Numeric(6,0) |
| EmployeeID | Numeric(9,0) |
| Room\_number | Numeric(4,0) |
| Check\_in\_date | MM/DD/YYYY, STRING = 10 CHARS |
| Key\_type | String<=20 |
| Check\_in\_time | hh-mm-ss, string = 8 chars |
| Lounge | Boolean |
| Length\_of\_stay | Integer>0 |

|  |  |  |
| --- | --- | --- |
| Relation Name | Attributes | Data Type |
| Bill\_for\_event | CustomerID | Numeric(6,0) |
| Event\_bill\_ID | Numeric(6,0) |
| Event\_ID | Numeric(4,0) |
| EmployeeID | Numeric(9,0) |
| Bill\_event\_date\_issued | MM/DD/YYYY, STRING = 10 CHARS |
| Bill\_event\_amount | Numeric(8,2) |

5 All Requested SQL Statements

5.1 Creation of Database with SQL Statements

5.1.1 Table Creation

--1

create table employee

(employeeid numeric(9,0) primary key,

age number(3) not null,

salary\_rate numeric(9,2),

name varchar2(20),

state varchar2(2),

city varchar2(20),

street\_name varchar2(20),

street\_number varchar2(20),

zipcode varchar2(10),

event\_staff\_id numeric(9,0) null

)

alter table employee add constraint fk\_employee

foreign key (event\_staff\_id) references employee (employeeid);

--2

create table dinning

(shift varchar2(10) null,

type varchar2(20) null,

employeeid numeric(9,0) primary key

)

alter table dinning add constraint fk\_dinning

foreign key (employeeid) references employee (employeeid);

--3

create table concierge

(

concierge\_experience\_year number(3) null,

employeeid numeric(9,0) primary key

)

alter table concierge add constraint fk\_concierge

foreign key (employeeid) references employee (employeeid);

--4

create table housekeeping

(

housekeeping\_experience\_year number(3) null,

employeeid numeric(9,0) primary key

)

alter table housekeeping add constraint fk\_housekeeping

foreign key (employeeid) references employee (employeeid);

--5

create table receptionist

(

language varchar2(10) null,

employeeid numeric(9,0) primary key

)

alter table receptionist add constraint fk\_receptionist

foreign key (employeeid) references employee (employeeid);

--6

create table tech\_support

(

tech\_support\_licence varchar2(10) null,

employeeid numeric(9,0) primary key

)

alter table tech\_support add constraint fk\_tech

foreign key (employeeid) references employee (employeeid);

--7

create table management

(

title varchar2(20) null,

employeeid numeric(9,0) primary key

)

alter table management add constraint fk\_management

foreign key (employeeid) references employee (employeeid);

--8

create table catering

(

employeeid numeric(9,0) primary key

)

alter table catering add constraint fk\_catering

foreign key (employeeid) references employee (employeeid);

--9

create table accountant

(

accountant\_licence varchar2(20) null,

employeeid numeric(9,0) primary key

)

alter table accountant add constraint fk\_accountant

foreign key (employeeid) references employee (employeeid);

--10

create table event

(

event\_id numeric(4,0) primary key,

event\_date date,

event\_time timestamp(6) with time zone,

event\_manager\_id numeric(9,0) not null,

event\_deposit numeric(7,2)

)

alter table event add constraint fk\_event

foreign key (event\_manager\_ID) references employee (employeeid);

-- 11. Client

CREATE TABLE client

(

customerid NUMERIC(6,0) PRIMARY KEY

);

-- 12. Organization

CREATE TABLE organization

(

customerid NUMERIC(6,0) PRIMARY KEY

);

ALTER TABLE organization add CONSTRAINT fk\_organization

FOREIGN KEY (customerid) REFERENCES client (customerid);

-- 13. Individual

CREATE TABLE individual

(

customerid NUMERIC(6,0) PRIMARY KEY,

birthdate DATE NULL,

individual\_name VARCHAR2(20) NULL,

individual\_sex VARCHAR2(20) NULL

);

ALTER TABLE individual add CONSTRAINT fk\_individual

FOREIGN KEY (customerid) REFERENCES client (customerid);

-- 14. Membersip

CREATE TABLE membership

(

membership\_number NUMERIC(11,0) PRIMARY KEY

);

-- 15. Room

CREATE TABLE room

(

room\_number NUMERIC(4,0) PRIMARY KEY,

bed\_type VARCHAR2(20) NULL,

room\_type VARCHAR2(20) NULL,

per\_night\_price INTEGER NULL

);

-- 16. Clear

CREATE TABLE clear

(

employeeid NUMERIC(9,0) NOT NULL,

room\_number NUMERIC(4,0) NOT NULL,

cdate DATE NULL,

ctime TIMESTAMP(6) WITH TIME ZONE,

PRIMARY KEY (employeeid, room\_number)

);

ALTER TABLE clear add CONSTRAINT fk\_clear1

FOREIGN KEY (employeeid) REFERENCES employee (employeeid);

ALTER TABLE clear add CONSTRAINT fk\_clear2

FOREIGN KEY (room\_number) REFERENCES room (room\_number);

-- 17. check\_out\_bill

CREATE TABLE check\_out\_bill

(

room\_number NUMERIC(6,0) NOT NULL,

customerid NUMERIC(6,0) NOT NULL,

check\_out\_bill\_amount INTEGER NOT NULL,

check\_out\_bill\_issed\_date DATE NOT NULL,

check\_in\_date DATE NOT NULL,

check\_out\_date DATE NOT NULL,

PRIMARY KEY (room\_number, customerid)

);

-- 18. has

CREATE TABLE has

(

customerid NUMERIC(6,0) NOT NULL,

membership\_number NUMERIC(11,0) NOT NULL,

PRIMARY KEY (customerid, membership\_number)

);

ALTER TABLE has add CONSTRAINT fk\_has1

FOREIGN KEY (customerid) REFERENCES client (customerid);

ALTER TABLE has add CONSTRAINT fk\_has2

FOREIGN KEY (membership\_number) REFERENCES membership (membership\_number);

--19

create table holds

(

customerid number(6,0) not null,

event\_id number(4,0) not null,

primary key (customerid, event\_id)

)

alter table holds add constraint fk\_holds

foreign key (customerid) references Client(customerid);

alter table holds add constraint fk\_holds2

foreign key (event\_id) references event (event\_id);

--20

create table serve

(

event\_id number(4,0) not null,

employeeid number(9,0) not null,

oncall\_speaker char(20) null,

oncall\_number number(4,0),

primary key (event\_id, employeeid)

)

alter table serve add constraint fk\_serve1

foreign key (employeeid) references employee (employeeid);

alter table serve add constraint fk\_serve2

foreign key (event\_id) references event (event\_id);

//////////////////////////////////////////

--21

create table bill\_event\_payment

(

customerid numeric(6,0),

event\_bill\_id numeric(6,0),

pdate date,

ptime TIMESTAMP(6) WITH TIME ZONE,

date\_of\_payment date,

amount\_of\_event\_payment numeric(8,2),

primary key (customerid, event\_bill\_id)

)

alter table bill\_event\_payment add constraint fk\_bill\_event\_payment

foreign key (customerid) references organization (customerid);

-- 22. direct\_bill\_account

CREATE TABLE direct\_bill\_account

(

customerid NUMERIC(6,0) NOT NULL,

direct\_bill\_account VARCHAR2(20) NOT NULL,

PRIMARY KEY (customerid, direct\_bill\_account)

);

ALTER TABLE direct\_bill\_account add CONSTRAINT fk\_direct\_bill\_account

FOREIGN KEY (customerid) REFERENCES client (customerid);

-- 23. phone

CREATE TABLE phone

(

customerid NUMERIC(6,0) PRIMARY KEY,

phone NUMERIC(10,0) NULL

)

ALTER TABLE phone add CONSTRAINT fk\_phone

FOREIGN KEY (customerid) REFERENCES client (customerid);

-- 24. check\_out\_payment\_id

CREATE TABLE check\_out\_payment\_id

(

customerid NUMERIC(6,0) NOT NULL,

room\_number NUMERIC(4,0) NOT NULL,

cop\_time TIMESTAMP(6) WITH TIME ZONE,

cop\_date DATE NOT NULL,

cop\_amount\_of\_payment NUMERIC(8,2),

PRIMARY KEY (customerid, room\_number)

);

ALTER TABLE check\_out\_payment\_id add CONSTRAINT fk\_check\_out\_payment\_id1

FOREIGN KEY (customerid) REFERENCES client (customerid);

ALTER TABLE check\_out\_payment\_id add CONSTRAINT fk\_check\_out\_payment\_id2

FOREIGN KEY (room\_number) REFERENCES room (room\_number);

--25

create table check\_in

(

customerid numeric(6,0) not null,

employeeid numeric(9,0) not null,

room\_number numeric(4,0) not null,

check\_in\_date date not null,

key\_type char(20),

check\_in\_time TIMESTAMP WITH LOCAL TIME ZONE not null,

length\_of\_stay numeric(9,0),

lounge numeric(1,0) default '0',

primary key (customerid, employeeid, room\_number, check\_in\_date, check\_in\_time)

)

alter table check\_in add constraint fk\_check\_in1

foreign key (employeeid) references receptionist(employeeid);

alter table check\_in add constraint fk\_check\_in2

foreign key (customerid) references client(customerid);

alter table check\_in add constraint fk\_check\_in3

foreign key (room\_number) references room(room\_number);

--26

create table bill\_for\_event

(

customerid number(6,0) not null,

event\_bill\_id number(6,0),

event\_id number(4,0) not null,

employeeid number(9,0) not null,

bill\_event\_date\_issued date,

bill\_event\_amount number(8,2),

primary key (customerid, event\_bill\_id)

)

alter table bill\_for\_event add constraint fk\_bill\_for\_event1

foreign key (employeeid) references accountant(employeeid);

alter table bill\_for\_event add constraint fk\_bill\_for\_event2

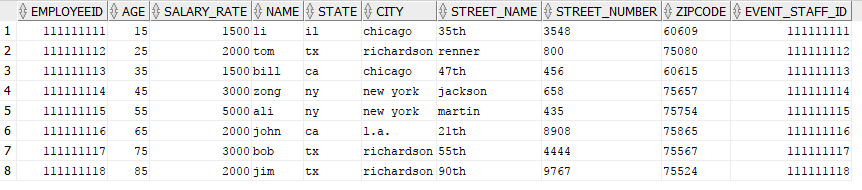
foreign key (customerid) references organization (customerid);

alter table bill\_for\_event add constraint fk\_bill\_for\_event3

foreign key (event\_id) references event (event\_id);

5.1.2 A Database State

--1. employee



--2. dinning



--3. concierge



--4. housekeeping



--5. receptionist



--6. tech\_support



--7. management



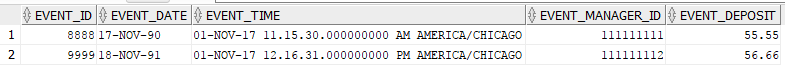
--8. catering



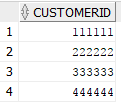
--9. accountant



--10. event



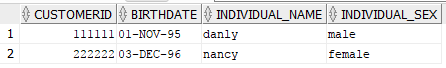
-- 11. client



-- 12. Organization



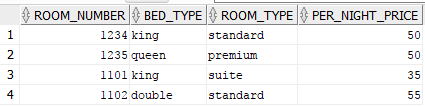
-- 13. Individual



-- 14. Membersip



-- 15. Room



-- 16. Clear



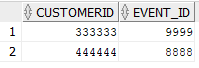
-- 17. check\_out\_bill



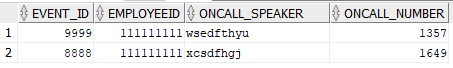
-- 18. has



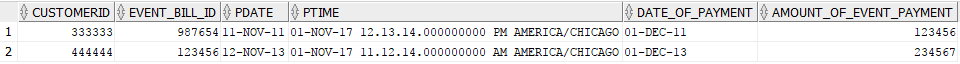
--19. holds



--20. serve



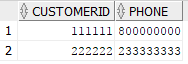
--21. bill\_event\_payment



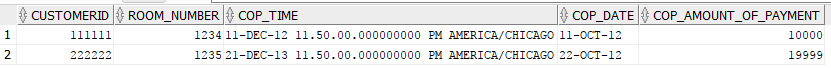
-- 22. direct\_bill\_account



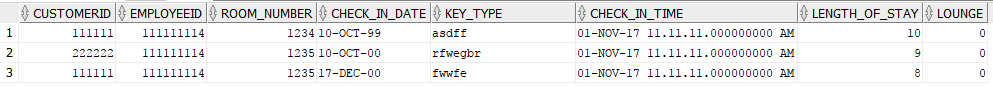
-- 23. phone



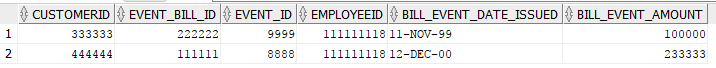
-- 24. check\_out\_payment\_id



--25. check\_in



--26. bill\_for\_event



5.2 Creation of Views (Answer for Question d/Phase III)

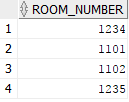
1.

CREATE VIEW available\_room (room\_number) AS

SELECT r.room\_number

FROM room r, check\_in ci

WHERE r.room\_number != ci.room\_number



2.

CREATE VIEW popular\_event\_manager (event\_staff\_id) AS

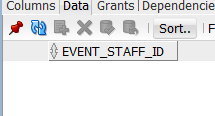
select event\_manager\_id

from event

where TO\_CHAR(event\_date, 'MMDD') >= TO\_CHAR(SYSDATE - 30, 'MMDD')

group by event\_manager\_id

having count(event\_id) > 10



3.

CREATE VIEW frequent\_customer (customerID) AS

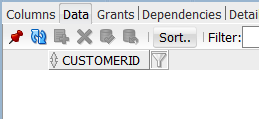
SELECT i.customerID

FROM individual i, check\_in ci

WHERE trunc(ci.Check\_in\_date, 'YEAR') = trunc(sysdate,'YEAR') and i.customerID = ci.customerID

GROUP BY i.customerID

having COUNT(ci.Check\_in\_date) >= 10



4.

CREATE VIEW popular\_room (room\_number) AS

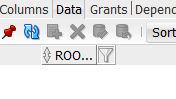
SELECT r.room\_number

FROM check\_in ci, room r

WHERE r.room\_number = ci.room\_number and trunc(ci.check\_in\_date, 'YEAR') > trunc(sysdate,'YEAR')

GROUP BY r.room\_number

having count (ci.check\_in\_date) > 30



5.3 Creation of SQL Queries (Answer for Question e/Phase III)

--1

select Count(employeeid) as nofEmployees

from dinning

where type = 'lounge' or type = 'bar'



-- 2

SELECT AVG(Salary\_rate)

FROM employee

WHERE employee.employeeid IN

(SELECT employeeid

FROM receptionist)



--3

select \*

from individual i

where i.CUSTOMERID in (

select customerid

from (

select c.customerid, Sum(c.COP\_AMOUNT\_OF\_PAYMENT)

from check\_out\_payment\_id c

where TRUNC(c.cop\_date, 'YEAR') = TRUNC(SYSDATE, 'YEAR')

group by c.customerid

having Sum(c.COP\_AMOUNT\_OF\_PAYMENT) > 1000

))

Result: empty

-- 4

SELECT Check\_out\_bill\_amount

FROM Check\_out\_bill

ORDER BY Check\_in\_date ASC



--5

select \*

from individual i

where i.CUSTOMERID in (

select customerid

from (

select c.customerid, Sum(c.length\_of\_stay)

from check\_in c

where TRUNC(c.check\_in\_date, 'YEAR') = TRUNC(SYSDATE, 'YEAR')

group by c.customerid

having Sum(c.length\_of\_stay) >= 15

)

);

Result: empty

-- 6

select avg(k.AGE\_TODAY)

from

(

select d.customerid, TRUNC((SYSDATE - TO\_DATE(d.bd, 'YYYY-MM-DD'))/ 365.25) AS AGE\_TODAY

from(

select birthdate as bd, i.customerid

from individual i

where i.customerid in

(

select customerid

from check\_in c

where c.employeeid in

(

select r.employeeid

from receptionist r

where r.language = 'Spanish'

))

) d

) k



--7

select \*

from organization o

where o.customerid in

(

select h.customerid

from holds h

where h.customerid in

(

select b.customerid

from Bill\_event\_payment b

group by b.CUSTOMERID

having Sum(b.amount\_of\_event\_payment) > 2000

)

group by h.customerid

having Count(Event\_id) >= 2

);

Result: empty

-- 8

select \*

from(

select b.bill\_event\_amount

from bill\_for\_event b

where b.event\_id in

(

select event\_id

from POPULAR\_EVENT\_MANAGER, event

)

order by b.bill\_event\_amount desc

)

where rownum = 1

Result: empty

--9

select \*

from event e

where e.event\_id in (

select event\_id

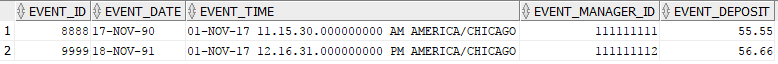
from(

select b.event\_id, b.customerid, max(bill\_event\_amount)

from bill\_for\_event b

group by b.event\_id, b.customerid

))



-- 10

select \*

from(

select c.check\_in\_date

from

(

select \*

from

(

select e.room\_number, Count(e.room\_number) as shax

from check\_in e

group by e.room\_number

order by shax desc

)

where rownum = 1

) d, check\_in c

where d.room\_number = c.room\_number

order by c.check\_in\_date desc

)

where rownum = 1



6. Dependency Diagram