



Museum Database Management System

By: Raghad Baroud 2110945
Bushra Alzeghabi 2110066
Bushra Abdulkhaleg 2110054

Instructor:
Amatulrahman Alharbi
Introduction to database
Course number: 10606
Section: A1

Description

A data base management system is designed to keep all the necessary information about a museum. a single museum contains galleries and each gallery has its own theme, staff, paintings and so on. A visitor makes a reservation to a specific gallery. Each gallery has it is own staff who guides the visitors in their journey and responsible for the safety of paintings. An artist has many paintings that could by displayed at the gallery. the building of this data base is necessary to solve many problems related to managing the museum and keep the data of visitors, so the museum could send coupons and Ad text. Our goal is to build a database that covers all the information about a specific museum.

Entities

Gallery: contains the gallery id, name, address, and size this information is to identify each gallery.

Staff: has ids, names, phone, and emails to communicate with them and a single staff can have many skills

Staff_skill: shows the skills of each employee.

Painting: is identified by a number has a size.

Artist: identified by id, has a name, email, phone, and each artist has a distinct type of art.

Reservation: has a number, payment method and the date of reservation.

Visitor: every visitor to the museum is identified uniquely by an id has a name, email, and phone number.

Description of the problem

These days, we are witnessing a lack of visitors in museums, one of the reasons for which is the lack of a clear database that contains a list of museums and their fields.

Visiting museums constantly enriches discussions, teamwork and sharing ideas among the participants, and the experience that visitors get from museums makes them comprehensive in understanding the various aspects of life. Therefore, exploring museums provides visitors with the willingness to participate enthusiastically in the process of acquiring information beyond their realm of knowledge. To make this list easier for visitors or even tourists, we will create a database of museums to make this process easier and easier.

We need this information the name of the museum, its type, in which gallery it will be displayed, and reservation information such as the number of people who will visit it, and the time and date.

Scenario description

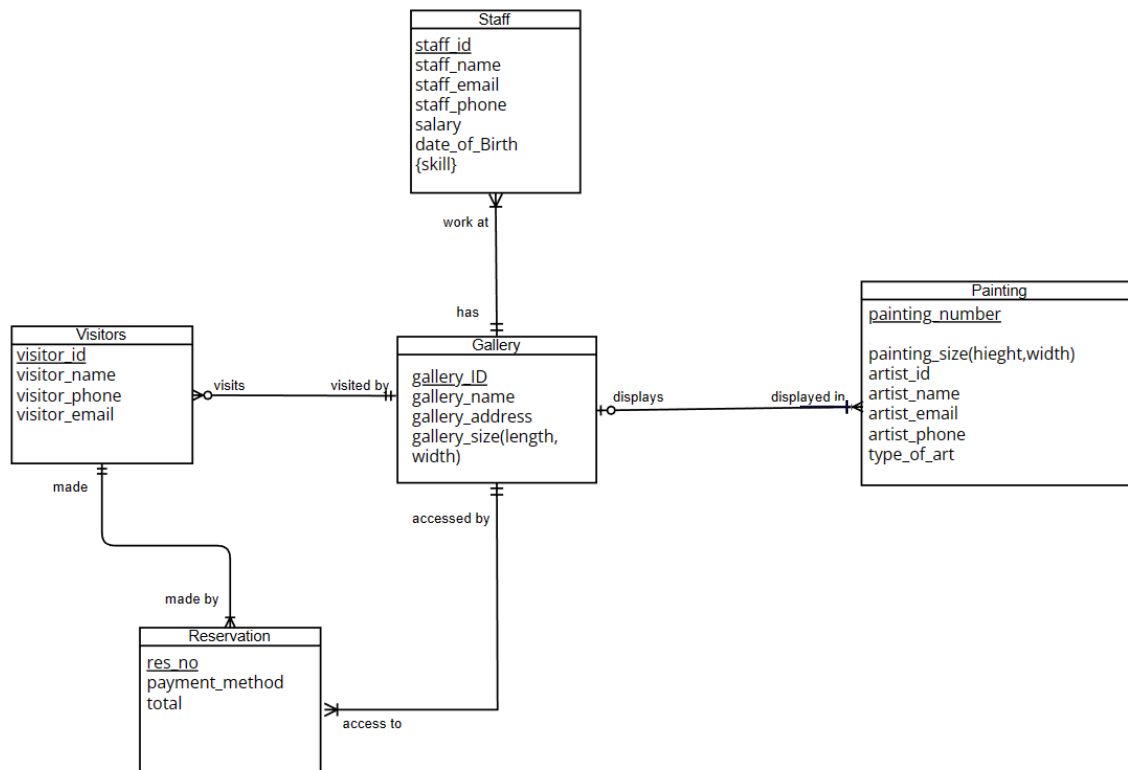
- Many visitors can visit the museum by making a reservation.
- The reservation is necessary for the visitors to enter the museum.
- Reservations of visitors including all the information such as: number of reservations, payment method, and total are kept in records.
- A single gallery may display many paintings.
- Painting owned by only one artist.
- Artist can have many Paintings.
- Each gallery has many staff works in it.
- A single staff could have many skills.
- Each gallery has name, id, address, and space.

Entity Table

Entities	Attributes
gallery	Gallery_id#, gallery_name, address, size(length, width).
artist	Artist_id#, artist_nama, artist_email, artist_phone, type_of_art.
painting	Painting_number#, width, height.
staff	Staff_id#, staff_name, staff_email, date_of_birth, staff_phone, sallery.
Staff_skill	Staff_id#, skill
visitor	Visitor_id#, visitor_name, visitor_phone, visitor_emial.
reservation	Rev_no#, payment_method, total.

Schemas Before Normalization

E-R Diagram



Relational Schema Diagram

painting

<u>Painting_number</u>	gallery_id	height	width	artist_id	artist_name	artist_email	artist_phone	type_of_art
------------------------	------------	--------	-------	-----------	-------------	--------------	--------------	-------------

gallery

<u>gallery_id</u>	gallery_name	gallery_address	length	width
-------------------	--------------	-----------------	--------	-------

staff

<u>staff_id</u>	<u>Gallery_id</u>	staff_name	Staff_email	Date_of_birth	salary	Staff_phone
-----------------	-------------------	------------	-------------	---------------	--------	-------------

staff_skill

<u>Staff_id</u>	<u>skill</u>
-----------------	--------------

visitor

<u>Visitor_id</u>	<u>Gallery_id</u>	Vistor_name	Visitor_phone	Visitor_email
-------------------	-------------------	-------------	---------------	---------------

reservation

<u>Rev_no</u>	<u>Gallery_id</u>	<u>Visitor_id</u>	Payment_method	total
---------------	-------------------	-------------------	----------------	-------

Functional Dependency

gallery_id#, -> gallery_name, gallery_address, length, width

staff_id# -> staff_name, date_of_birth, salary, staff phone, staff_email

visitor_id# -> visitor_name, visitor_phone, visitor_email

staff_id#, skill# -> skill

res_no# -> payment_method, total

painting_number# -> height, width, artist_id#, artist_name, artist_phone, artist_email, type_of_art

Normalization Process

Normalization Using Rational Scheme

All relations are in second normal form because all the non-key attributes are fully dependent on the primary key of a relation (2NF).

Anomalies

Insertion anomalies: a new artist cannot be inserted unless he has a painting in the gallery.

Deletion anomalies: if a painting is deleted the whole information of the artist is deleted too.

Update anomalies: if an artist information is updated, all paintings of that artist must be updated to (not atomic).

The relation has transitive dependency. Thus, the relation is not in third normal form (3NF).

painting

<u>Painting_number</u>	gallery_id	height	width	artist_id	artist_name	artist_email	artist_phone	type_of_art
------------------------	------------	--------	-------	-----------	-------------	--------------	--------------	-------------

Painting (painting_number#, gallery_id, painting_name, height, width, artist_id, artist_email, artist_phone, type_of_art).

We must create a new relation for all the non-keys attributes that depends on another non-key attribute.

painting

<u>Painting_number</u>	<u>Gallery_id</u>	<u>Artist_id</u>	height	width
------------------------	-------------------	------------------	--------	-------

Artist

<u>Artist_id</u>	Artist_name	Artist_phone	Type_of_art	Artist_email
------------------	-------------	--------------	-------------	--------------

Painting (painting_number#, gallery_id#, artist_id#, painting_name, height, width).

Artist (artist_id#, artist_name, artist_phone, type_of_art).

All the other relations do not have transitive dependency.
Thus, all in third normal form and in Boyce normal form.

Galley (gallery_id#, gallery_name, gallery_address, length, size)

Staff (staff_id#, gallery_id#, staff_name, date_of_birth, salary, staff phone)

Staff_skill (staff_id#, skill#)

Visitor (visitor_id#, gallery_id#, visitor_name, visitor_phone, visitor_email)

Reservation (res_no#, gallery_id#, visitor_id#, payment_method, total)

Painting (painting_number#, artist_id#, gallery_id#, height, width)

Artist (artist_id#, artist_name, artist_phone, artist_email, type_of_art)

Artist

<u>artist_id</u>	artist_name	artist_email	artist_phone	type_of_art
------------------	-------------	--------------	--------------	-------------

Painting

<u>Painting_number</u>	gallery_id	artist_id	width	height
------------------------	------------	-----------	-------	--------

gallery

<u>gallery_id</u>	gallery_name	gallery_address	length	width
-------------------	--------------	-----------------	--------	-------

staff

<u>staff_id</u>	<u>Gallery_id</u>	staff_name	Staff_email	Date_of_Birth	salary	Staff_phone
-----------------	-------------------	------------	-------------	---------------	--------	-------------

staff_skill

<u>Staff_id</u>	<u>skill</u>
-----------------	--------------

visitor

<u>Visitor_id</u>	<u>Gallery_id</u>	Vistor_name	Vistor_phone	Visitor_email
-------------------	-------------------	-------------	--------------	---------------

reservation

<u>Rev_no</u>	<u>Gallery_id</u>	<u>Visitor_id</u>	Payment_method	total
---------------	-------------------	-------------------	----------------	-------

Normalization Using User View

Form containing information for a specific gallery.

Gallery_id:

Gallery_name:

Gallery_address:

Gallery_size (length, width):

Visitor_id#	Visitor_name	Visitor_email	Visitor_phone

Res_no#	Payment_method	total

Staff_id#	Staff_name	Staff_email	Staff_phone	salary	Date_of_Birth	skill

Painting_number#	height	width	Artist_id#	Artist_name	Artist_phone	Type_of_art	artist_email

Gallery (gallery_id#, gallery_name, gallery_address, length, width (visitor_id#, visitor_email, visitor_phone(res_no#, payment_method ,date_of_birth, salary, total)), (res_no#, payment_method ,date_of_birth, salary, total), (staff_id#, staff_name, staff_phone,{ skill }), (painting_number#, height, width, artist_id, artist_name, artist_phone, artist_email, type_of_art))

Currently in un-normalized form 'UNF'

to convert the relation into first normal form we remove the repeating groups by creating new relations and each relation must have a primary key.

First, classifying the repeating groups

each gallery has many visitors and reservations, many people work at the gallery known as staff those staff could have more than one skills. The gallery displays many paintings the paintings made by one and only one artist. A visitor can make many reservations.

Knowing the above repeating groups now we create relation to each of them and define a primary key at this point it does not matter if the attribute in a relation is partially or fully dependent on the primary key.

... 1NF

Gallery (gallery_id#, gallery_name, gallery_address, length, width)

Visitor (visitor_id#, gallery_id#, visitor_name, visitor_email, visitor_phone)

Reservation (res_no#, gallery_id#, visitor_id#, payment_method, total)

Staff (staff_id#, gallery_id#, staff_name, staff_email, date_of_birth, salary, staff_phone)

Staff_skill (staff_id#, skill)

Painting (painting_number#, gallery_id#, artist_id#, height, width, artist_name, artist_email, artist_phone, type_of_art)

Now, the relation is in 1NF and in 2NF because all the non-key attributes fully dependents on the primary key.

Painting relation is not in 3NF.

Anomalies

Insertion anomalies: a new artist cannot be inserted unless he has a painting in the gallery.

Deletion anomalies: if a painting is deleted the whole information of an artist is deleted.

Update anomalies: if an artist's information is updated, all paintings of that artist must be updated to (not atomic).

The relation has transitive dependency. Thus, the relation is not in third normal form (3NF).

Painting (painting_number#, gallery_id#, artist_id#, height, width).

Artist (artist_id#, artist_name, artist_phone, artist_email, type_of_art).

All the other relations are in 3NF:

Galley (gallery_id#, gallery_name, gallery_address, length, width)

Staff (staff_id#, gallery_id#, staff_name, date_of_birth, salary, staff_phone)

Staff_skill (staff_id#, skill#)

Visitor (visitor_id#, gallery_id#, visitor_name, visitor_phone, visitor_email)

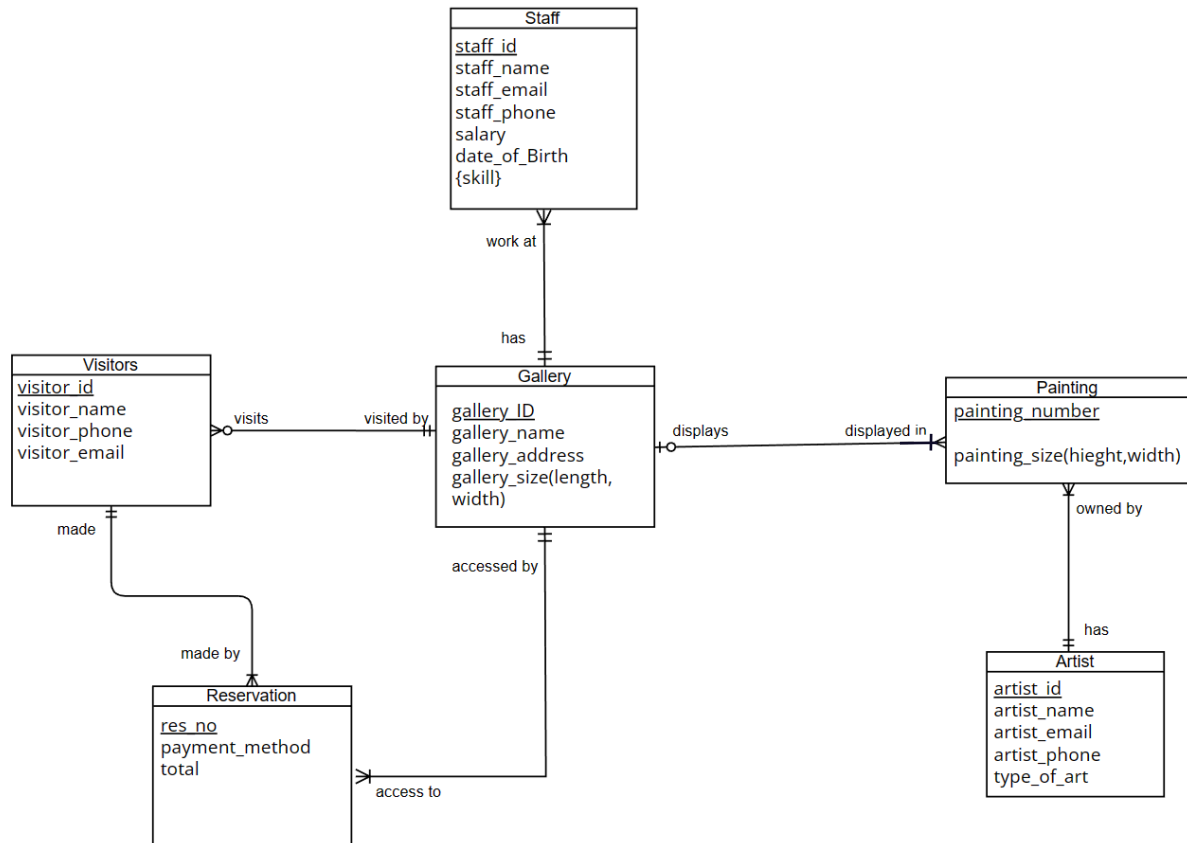
Reservation (res_no#, gallery_id#, visitor_id#, payment_method, total)

Painting (painting_number#, artist_id#, gallery_id#, height, width)

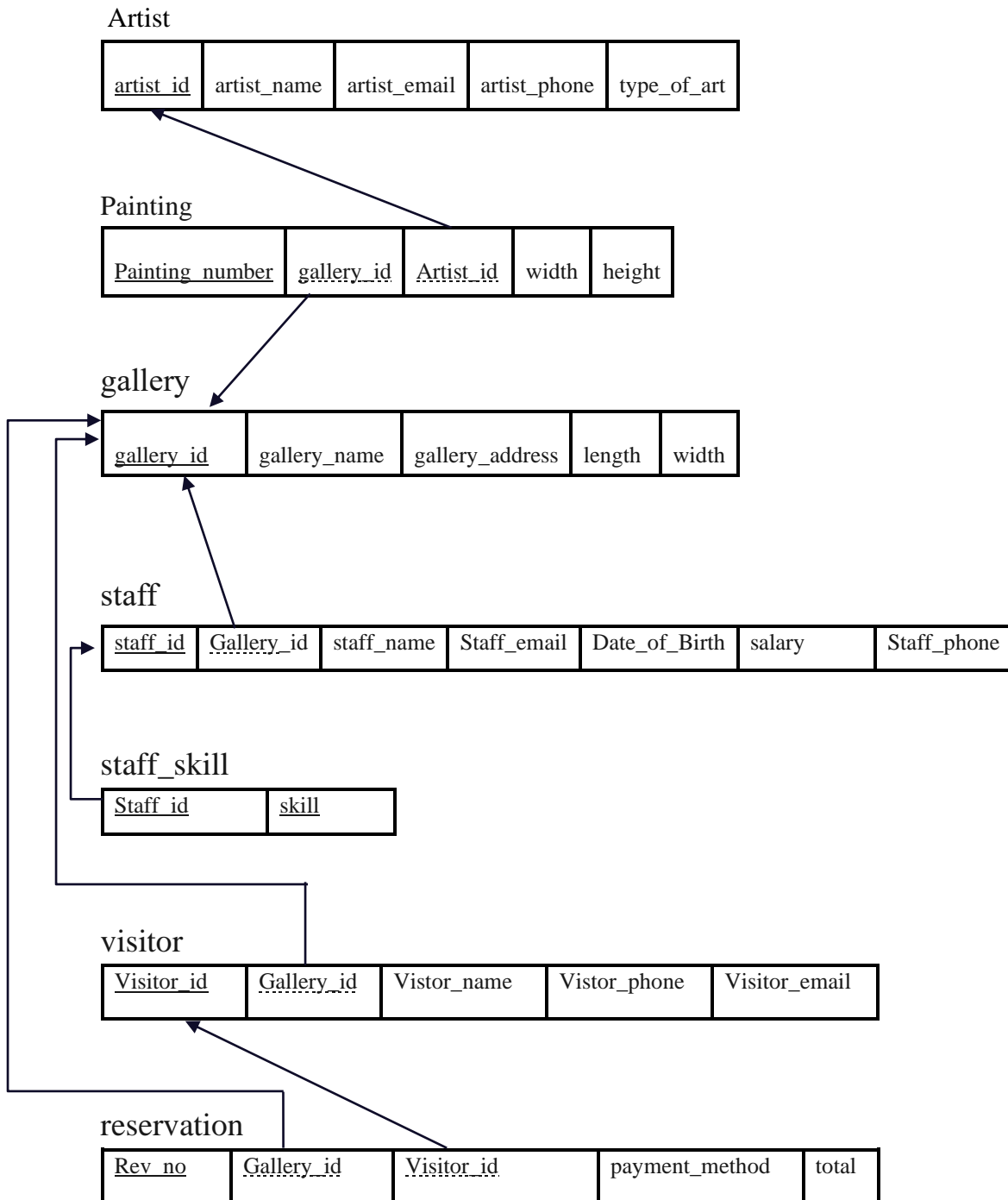
Artist (artist_id#, artist_name, artist_email, artist_phone, type_of_art)

Schemas After Normalization

E-R Diagram



Relational Schema Diagram



Functional Dependency

gallery_id# -> gallery_name, gallery_address, length, width

staff_id# -> staff_name, date_of_birth, salary, staff phone, staff_email

visitor_id# -> visitor_name, visitor_phone, visitor_email

staff_id#, skill# -> skill

res_no# -> payment_method, total

painting_number# -> height, width

artist_id# -> artist_name, artist_phone, artist_email, type_of_art

Oracle Screenshots

Create Relations

1

SQL Worksheet

```
1 create table reservation
2 (
3     res_no number(4),
4     gallery_id number(2),
5     visitor_id number(3),
6     payment_method varchar2(30),
7     total number(4)
8 );
9 create table visitor
10 (
11     visitor_id number(3),
12     gallery_id number(2),
13     vname varchar2(20),
14     vemail varchar2(50) unique,
15     vphone varchar2(10) not null
16 );
17
```

Table created.

Table created.

2

SQL Worksheet

```
16 );
17
18 create table staff
19 (
20     staff_id number(5),
21     gallery_id number(2),
22     sname varchar2(30),
23     salary number(6),
24     date_of_Birth date,
25     semail varchar2(50) unique,
26     sphone varchar2(10) not null
27 );
28
29 create table staff_skill
30 (
31     staff_id number(5),
```

Table created.

3

SQL Worksheet

```
28
29 create table staff_skill
30 (
31     staff_id number(5),
32     skill varchar2(50)
33 );
34
35 create table gallery
36 (
37     gallery_id number(2),
38     gname varchar2(20),
39     address varchar2(30),
40     glength varchar2(10),
41     width varchar2(10)
42 );
43
```

Table created.

Table created.

4

SQL Worksheet

```
45 create table painting
46 (
47     pnumber number(2),
48     gallery_id number(2),
49     artist_id number(4),
50     height number(10),
51     width number(10)
52 );
53
54 create table artist
55 (
56     artist_id number(4),
57     artist_name varchar2(30),
58     artist_email varchar2(50) not null,
59     aphone varchar2(10) unique,
60     type_of_art varchar2(30)
61 );
```

Table created.

Table created.

Primary key

1

SQL Worksheet

```
60 type_or_art varchar(30)
61 );
62
63 alter table reservation
64 add constraint res_pk primary key(res_no);
65
66 alter table visitor
67 add constraint vis_pk primary key(visitor_id);
68
69 alter table staff
70 add constraint staff_pk primary key(staff_id);
71
72 alter table staff_skill
73 add constraint skill_pk primary key(staff_id,skill);
74
75 alter table gallery
```

Table altered.

Table altered.

Table altered.

Table altered.

2

SQL Worksheet

```
74
75 alter table gallery
76 add constraint gall_pk primary key(gallery_id);
77
78 alter table painting
79 add constraint paint_pk primary key(pnumber);
80
81 alter table artist
82 add constraint art_pk primary key(artist_id);
83
```

Table altered.

Table altered.

Table altered.

Foreign key

```
83
84
85 alter table reservation
86 add constraint res_vis_fk foreign key(visitor_id) references visitor(visitor_id) on delete cascade;
87
88 alter table reservation
89 add constraint res_gall_fk foreign key(gallery_id) references gallery(gallery_id);
90
91 alter table visitor
92 add constraint vis_gal_fk foreign key(gallery_id) references gallery(gallery_id);
93
94 alter table staff
```

Table altered.

Table altered.

Table altered.

```
92 add constraint vis_gal_fk foreign key(gallery_id) references gallery(gallery_id);
93
94 alter table staff
95 add constraint staff_gal_fk foreign key(gallery_id) references gallery(gallery_id);
96
97 alter table staff_skill
98 add constraint skill_staff_fk foreign key(staff_id) references staff(staff_id);
99
100 alter table painting
101 add constraint paint_arti_fk foreign key(artist_id) references artist(artist_id);
102
103
```

Table altered.

Table altered.

Table altered.

Insert Information

```
102
103
104
105 insert into gallery values(01,'Antiquities','North','600m','300m');
106 insert into gallery values(02,'Mural','North','600m','600m');
107 insert into gallery values(03,'Artvago','South','400m','300m');
108 insert into gallery values(04,'Clip Art','South','500m','500m');
109 insert into gallery values(05,'Craftrix','South','600m','500m');
110 insert into gallery values(06,'Piece of Work','East','800m','400m');
111 insert into gallery values(07,'Global Art','West','800m','400m');
112
113 insert into visitor values(100,01,'Bushra','bobo123@gmail.com','0558805088');
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

```
--- insert into gallery, values(01, 'Antiquities', 'North', '600m', '300m'),
112
113 insert into visitor values(100,01,'Bushra','bobo123@gmail.com','0558805088');
114 insert into visitor values(101,05,'Sara','S2002@gmail.com','0559994405');
115 insert into visitor values(102,07,'Ali','Ali@gmail.com','0554464598');
116 insert into visitor values(103,05,'Adeem','Adeem@gmail.com','0553456789');
117 insert into visitor values(104,03,'Noor','Noor@gmail.com','0552314589');
118 insert into visitor values(105,02,'Ghanem','Gh55@gmail.com','0558565755');
119 insert into visitor values(106,04,'Mohammad','Mohammad@gmail.com','0557788991')
120
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

```

120
121 insert into reservation values(1000,01,100,'Credit Card',60);
122 insert into reservation values(1001,03,101,'Apple pay',15);
123 insert into reservation values(1002,04,103,'cache',30);
124 insert into reservation values(1003,06,104,'Apple pay',75);
125 insert into reservation values(1004,06,101,'Credit Card',15);
126 insert into reservation values(1005,07,105,'Apple pay',30);
127
128 insert into staff values(00001,01,'Ahmed',10000,'28-SEP-22','Ahmed1
129 insert into staff values(00002,02,'Amir',50000,'20-FEB-21','Amir20@
130 insert into staff values(00003,03,'Khalil',15000,'19-SEP-20','Khali
131 insert into staff values(00004,04,'Nasir',320000,'28-FEB-23','Nasir

```

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

127

```

128 insert into staff values(00001,01,'Ahmed',10000,'28-SEP-22','Ahmed10@gmail.com','0551021314');
129 insert into staff values(00002,02,'Amir',50000,'20-FEB-21','Amir20@gmail.com','0557766889');
130 insert into staff values(00003,03,'Khalil',15000,'19-SEP-20','Khalil30@gmail.com','0556644894');
131 insert into staff values(00004,04,'Nasir',320000,'28-FEB-23','Nasir40@gmail.com','0556543278');
132 insert into staff values(00005,05,'Jawad',20000,'15-APR-21','Jawad50@gmail.com','0551456732');
133 insert into staff values(00006,06,'Salah',27000,'20-MAY-22','Salah60@gmail.com','0553457892');
134 insert into staff values(00007,07,'Zaid',30000,'01-MAY-21','Zaid70@gmail.com','0554327652');
135
136 insert into staff_skill values(00001,'Critical thinking');
137 insert into staff_skill values(00002,'Teamwork and collaboration');
138 insert into staff_skill values(00003,'Teamwork and collaboration');

```

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.


```

134 insert into staff values(00007,07,'Zaid',30000,'01-MAY-21','Zaid70@gmail.com','e
135
136 insert into staff_skill values(00001,'Critical thinking');
137 insert into staff_skill values(00002,'Teamwork and collaboration');
138 insert into staff_skill values(00003,'Teamwork and collaboration');
139 insert into staff_skill values(00004,'Leadership');
140 insert into staff_skill values(00005,'Professionalism and strong work ethic');
141 insert into staff_skill values(00006,'Teamwork and collaboration');
142 insert into staff_skill values(00007,'problem solving');
143
144 insert into artist values(100,'Amer','AmerARTIST@gmail.com','0554433665','collag
145 insert into artist values(200,'Asad','AsadARTIST@gmail.com','0557766546','Fine A

```

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

```

143
144 insert into artist values(100,'Amer','AmerARTIST@gmail.com','0554433665','collages');
145 insert into artist values(200,'Asad','AsadARTIST@gmail.com','0557766546','Fine Art');
146 insert into artist values(300,'Badr','BadrARTIST@gmail.com','0557894367','modern art');
147 insert into artist values(400,'Bilal','BilalARTIST@gmail.com','0553322424','modern art');
148 insert into artist values(500,'Faiz','FaizARTIST@gmail.com','0552234342','ancient art');
149 insert into artist values(600,'Hamza','HamzaARTIST@gmail.com','0551234467','decorative art');
150 insert into artist values(700,'Hazem','HazemARTIST@gmail.com','0554433656','collages');
151
152

```

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

```
152 insert into painting values(30,03,100,160,60);
153 insert into painting values(31,02,200,260,60);
154 insert into painting values(32,04,300,100,50);
155 insert into painting values(33,05,400,80,60);
156 insert into painting values(34,05,500,60,60);
157 insert into painting values(36,06,600,120,80);
158 insert into painting values(37,07,700,120,40);
159 insert into painting values(38,03,100,160,60);
160 insert into painting values(39,02,200,260,60);
161 insert into painting values(40,05,300,100,50);
162 insert into painting values(41,05,400,80,60);
163 insert into painting values(42,05,500,60,60);
164 insert into painting values(43,06,600,120,80);
165 insert into painting values(44,01,700,120,40);
```

```
1 row(s) inserted.
```

```
1 row(s) inserted.
```

```
1 row(s) inserted.
```

```
1 row(s) inserted.
```

```
1 row(s) inserted.
```

```
1 row(s) inserted.
```

Queries

1- using subquery to check the skill of staff member named jawad.

```
169 -- using subquery to check the skill of staff member named Jawad
170 select staff_id as id, skill
171 from staff_skill
172 where staff_id = (select staff_id
173                  from staff
174                  where sname = 'Jawad');
175
176
```

ID	SKILL
5	Professionalism and strong work ethic

Download CSV

2- Retrieving financial data in descending order.

```
177 -- retrieving financial data in descending order
178 select total, payment_method as "PAYMENT METHOD"
179 from reservation
180 order by total desc;
181
```

TOTAL	PAYMENT METHOD
75	Apple pay
60	Credit Card
30	Apple pay
30	cache
15	Credit Card
15	Apple pay

4- Print the name of visitors to each gallery.

```
102
183 -- print the name of visitors to each gallery
184 select g.gallery_id, g.gname, v.vname
185 from gallery g left outer join visitor v
186 on g.gallery_id = v.gallery_id
187 order by gallery_id asc;
```

GALLERY_ID	GNAME	VNAME
1	Antiquities	Bushra
2	Mural	Ghanem
3	Artvago	Noor
4	Clip Art	Mohammad
5	Craftric	Sara
5	Craftric	Adeem
6	Piece of Work	-
7	Global Art	Ali

5- Print the number of paintings each artist has.

```
190
191 -- print the number of paintings each artist have
192 select artist_id, count(pnumber) as "Number of Paintings"
193 from artist natural join painting
194 group by artist_id
195 order by artist_id; |
196
```

ARTIST_ID	Number of Paintings
100	2
200	2
300	2
400	2
500	2
600	2
700	2

6- Retrieve the names of the galleries with profit higher than the average

```
197 -- print the names of the galleries with profit heigher than the average
198 select gname as "Gallery Name", total
199 from gallery join reservation
200 on gallery.gallery_id = reservation.gallery_id
201 where total > all(select avg(total)
202                  from reservation)
203 order by total desc;
204
```

Gallery Name	TOTAL
Piece of Work	75
Antiquities	60

Download CSV

7- print phone number of visitors who paid using apple pay so we can send them coupon to their SMS.

```
206 -- print phone number of visitors who payed using
207 select vphone
208 from visitor join reservation
209 on visitor.visitor_id = reservation.visitor_id
210 where payment_method = 'Apple pay';
```

VPHONE
0559994405
0552314589
0558565755

Download CSV

8- Retrieve the minimum and maximum salary that this museum institution offers to staff.

```
212 -- retrieve the minimum and maximum salary that this musuem institution offers to staff.  
213 select max(salary) as "maximum salary",min(salary) as "minimum salary"  
214 from staff ;  
215  
216 -- procedures
```

maximum salary	minimum salary
320000	10000

Download CSV

Create Procedure

1.

show staff information and compare average salary for all staff in gallery and compare between user enter salary

```
218 --show staff information and compare average salary for all staff in gallery and compare between user enter salary
219 CREATE OR REPLACE PROCEDURE staff_sal(
220     p_staff IN staff.staff_id%TYPE
221 )
222 AS
223     gid          staff.gallery_id%TYPE;
224     s_name        staff.sname%TYPE;
225     s_salary      staff.salary%TYPE;
226     DB            staff.date_of_Birth%TYPE;
227     s_email       staff.semail%TYPE;
228     s_phone       staff.sphone%TYPE;
229     s_guess       s_salary%TYPE;
230 BEGIN
231     select gallery_id,sname,salary,date_of_Birth,semail,sphone
232     into gid,s_name,s_salary,DB,s_email,s_phone
233     from staff where staff_id = p_staff;
234     DBMS_OUTPUT.PUT_LINE('STAFF ID : ' || p_staff );
235     DBMS_OUTPUT.PUT_LINE('which gallery : ' || gid );
236     DBMS_OUTPUT.PUT_LINE('STAFF NAME : ' || s_name );
237     DBMS_OUTPUT.PUT_LINE('STAFF SALRAY : ' || s_salary );
238     DBMS_OUTPUT.PUT_LINE('STAFF BIRTH : ' || DB );
239     DBMS_OUTPUT.PUT_LINE('STAFF EMAIL : ' || s_email );
240     DBMS_OUTPUT.PUT_LINE('STAFF PHONE : ' || s_phone );
241
242     select AVG(salary) into s_guess
243     from staff where gallery_id=gid ;
244     if s_salary > s_guess then
245         DBMS_OUTPUT.PUT_LINE('Staff salay is greater than you enter:' ||s_guess);
246     else
247         DBMS_OUTPUT.PUT_LINE('Staff salay is lower than you enter:' ||s_guess);
248     end if;
249 end;
```

Procedure created.

```
248     end if;
249 end;
250
251 exec staff_sal (00002);
252
```

```
Statement processed.
STAFF ID :2
which gallery :2
STAFF NAME :Amir
STAFF SALRAY :50000
STAFF BIRTH :20-FEB-21
STAFF EMAIL :Amir20@gmail.com
STAFF PHONE :0557766889
Staff salay is lower than you enter:50000
```

2.

here to print all details about chosen painting

```
252
253 -- here to print all details about chosen painting
254 CREATE OR REPLACE PROCEDURE painting_info(p_id IN painting.pnumber%TYPE)
255 AS
256 BEGIN
257     FOR REC IN (SELECT gallery_id,artist_id,height,width FROM painting WHERE pnumber=p_id ) LOOP
258         DBMS_OUTPUT.PUT_LINE('which gallery: ' || REC.gallery_id || ' ' || 'which artist: ' || REC.artist_id || ' ' || 'height: ' || REC.height || ' ' || 'width: ' || REC.width || ' ');
259     end loop;
260 exception when NO_DATA_FOUND THEN NULL;
261 END;
262
263 exec painting_info(36);
264
265
266 --update procedure for visitor information
267 CREATE OR REPLACE PROCEDURE updatevisitor(
268     p_id IN visitor.visitor_id%TYPE,
269     p_name IN visitor.vname%TYPE,
270     p_email IN visitor.vemail%TYPE,
271     p_phone IN visitor.vphone%TYPE
```

Procedure created.

```
261 END;
262
263 exec painting_info(36);
264
265
```

Statement processed.
which gallery:6 which artist:600 height:120 width:80

3.

update procedure for visitor information

```
266 --update procedure for visitor information
267 CREATE OR REPLACE PROCEDURE updatevisitor(
268     p_id IN visitor.visitor_id%TYPE,
269     p_name IN visitor.vname%TYPE,
270     p_email IN visitor.vemail%TYPE,
271     p_phone IN visitor.vphone%TYPE
272 )
273 IS
274 BEGIN
275
276     UPDATE visitor SET vname = p_name,vemail=p_email,vphone=p_phone where visitor_id = p_id;
277
278     COMMIT;
279
280 END;
281
282 exec updatevisitor(100,'razan','razan@gmail.com',0541216773);
283 select * from visitor;
```

Procedure created.

```
280 END,
281
282 exec updatevisitor(100,'razan','razan@gmail.com',0541216773);
283 select * from visitor;
284
```

VISITOR_ID	GALLERY_ID	VNAME	VEMAIL	VPHONE
100	1	razan	razan@gmail.com	541216773
101	5	Sara	S2002@gmail.com	0559994405
102	7	Ali	Ali@gmail.com	0554464598
103	5	Adeem	Adeem@gmail.com	0553456789
104	3	Noor	Noor@gmail.com	0552314589
105	2	Ghanem	Gh55@gmail.com	0558565755
106	4	Mohammad	Mohammad@gmail.com	0557788991

4.

CURSOR to check for attributes and compare total with input user and display all matches.

```
288 -- CURSOR to check for attributes and compare total with input user and display all matches
289 CREATE OR REPLACE PROCEDURE totalR(evaluate number)
290 AS
291 CURSOR executive IS
292 select res_no, total
293 from reservation
294 where total > evaluate;
295 begin
296 for v_cursrec in executive loop
297 DBMS_OUTPUT.PUT_LINE(v_cursrec.res_no || ' ' || v_cursrec.total);
298 end loop;
299 end totalR;
```

Procedure created.

```
299 end totalR;
300
301 exec totalR(15);
302
```

Statement processed.

```
1000 60
1002 30
1003 75
1005 30
```

5.

users enter reservation number that want to show total with tax

```
303  --user enter reservation number that want to show total with tax
304  CREATE OR REPLACE PROCEDURE tax(p_id IN reservation.res_no%TYPE)
305  as
306  begin
307  FOR REC IN (SELECT total from reservation where res_no = p_id ) LOOP
308  DBMS_OUTPUT.PUT_LINE('final total with tax:'||REC.total*1.05);
309  End Loop;
310  End;
311
```

Procedure created.

```
311
312  exec tax(1001);
313
```

Statement processed.
final total with tax:15.75