



American International University- Bangladesh (AIUB)

Faculty of Science and Technology

CSC 2107: INTRODUCTION TO DATABASE

“ ART GALLERY MANAGEMENT SYSTEM ”

Supervisor

JUENA NOSHIN RAHMAN

Author

ABRAR, M. AUMLAN [17-34206-1]

Academic Year: 2017-2018

CONTENT LIST:

No 1	INTRODUCTION
No 2	SCENARIO
No 3	ER-DIAGRAM
No 4	NORMALIZATION
No 5	TABLE CREATION
No 6	DATA INSERTION
No 7	QUERY WRITING
No 8	CONCLUTION

INTRODUCTION

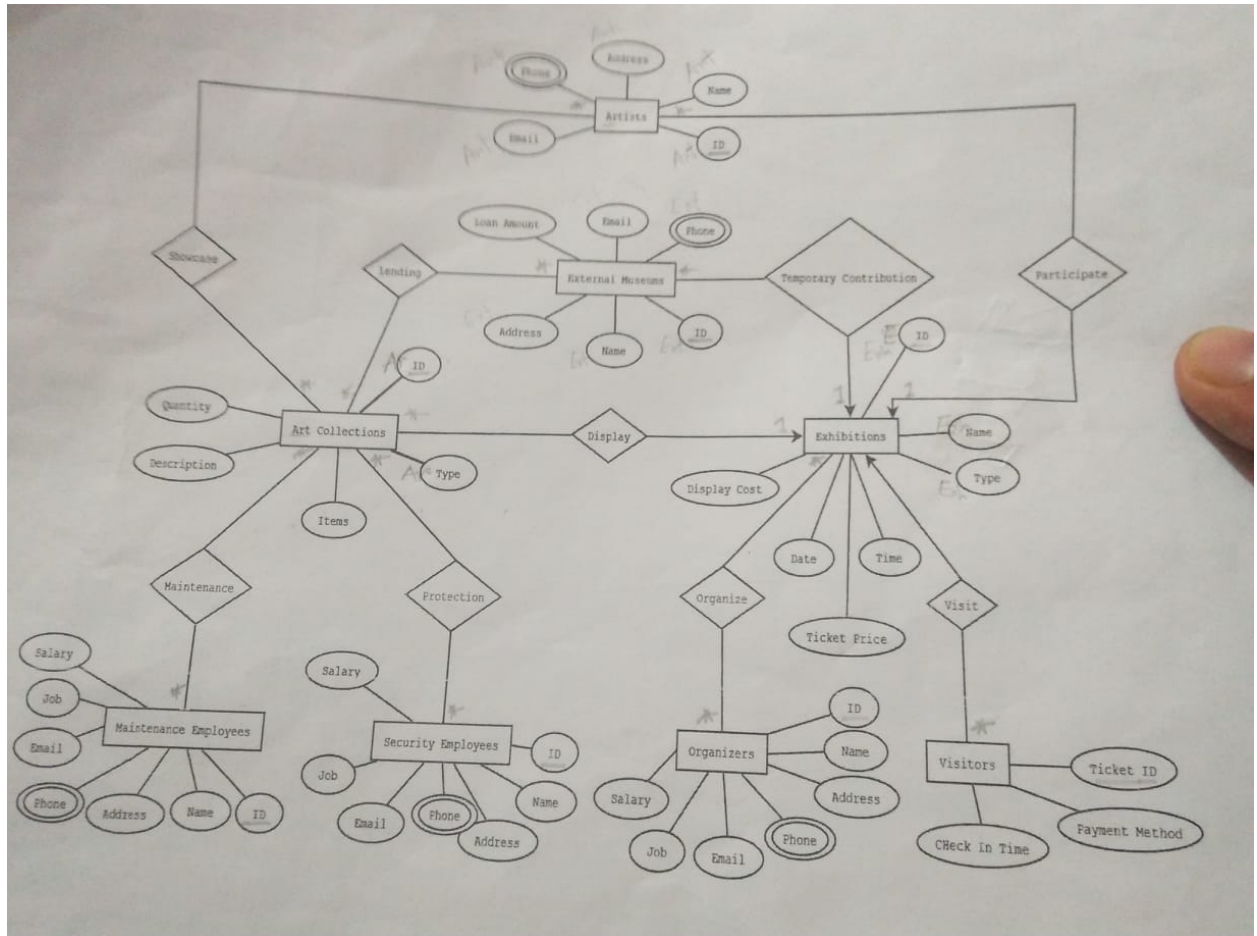
Art gallery is the place where the arts or art crafts of the artists can be displayed for the visitors view. People also comment on the arts that artist's exhibit so that there will be some scope of improvement of the arts that is displayed. Sometimes it will be difficult to maintain the details of the artists and their paintings details through the pen paper method.

The art gallery management is the system that allows the art gallery employees to maintain the details of the artists and the details of their paintings with great ease. This will be one of the useful applications for the owners of the art gallery which will reduce most of their work.

SCENARIO

In a art gallery system art collections are displayed in exhibition. An exhibition must have many art collections. Art collections have at least one exhibition. Art collection identified by ID. It also have quality, items, type and description. An exhibition is identified by ID. It also stored name, type. An exhibition also contains date, display cost and ticket price. Artist store their art collection in showcase. Artist may have many art collections. An artist is identified by his ID. It also stored name, email, address and phone number of the artists. External museums lend art collections from the system. While lending system stores external museums name, ID, email, address, loan amounts and phone numbers. External museums temporarily contributes in exhibitions. An art gallery system, External museums have temporary contribution in at least one exhibition. In an exhibition many artists may participate. Art collections are maintained by maintenance employees. Maintenance employees are identified by ID. They also have name, address, email and phone number. Phone number may be multi valued. Their job and salary are also stored in the system. Security employees protect art collection. While storing their job and salary, system also store their ID, name, email, phone number. Many organizers organize many exhibition. An organizer is identified by ID, name, email, address, phone number, job and salary. Many visitors visit at least one exhibition. Visitors are identified by visitors ID. The system also stored visitors name, payment method and opinions of the visitors.

ER-Diagram



NORMALIZATION

ART COLLECTIONS---*---DISPLAY---1---EXHIBITIONS

UNF:

(art_id, art_type, items, quality, description, exh-id, exh_name, exh_type, display_cost, date, time, ticket_price)

1NF:

There is no multivalued attribute.

(art_id, art_type, items, quality, description, exh_id, exh_name, exh_type, display_cost, date, time, ticket_price)

2NF:

1. art_id, art_type, items, quality, description.
2. exh_id, exh_name, exh_type, display_cost, date, time, ticket_price.

3NF:

1. art_id, art_type, items, quality, description.
2. exh_id, exh_name, exh_type, display_cost, date, time, ticket_price.

Table:

<u>art_id</u> , art_type, items, quality, description, <u>exh_id</u> .
<u>exh_id</u> , exh_name, exh_type, display_cost, date, time, ticket_price.

ART COLLECTIONS---*---LENDING---*--- EXTRENAL MUSEUMS

UNF:

(art_id, art_type, items, quality, description, ext_id, ext_address, ext_name, ext_e,mail, ext_phone, loan amount)

2NF:

Phone is a multivalued attribute.

(art_id, art_type, items, quality, description, ext_id, ext_address, ext_name, ext_e,mail, ext_phone, loan_amount)

2NF:

1. art_id, art_type, items, quality, description.
2. ext_id, ext_address, ext_name, ext_email, ext_phone, loan_amount.

3NF:

1. art_id, art_type, items, quality, description.
2. ext_id, ext_address, ext_name, ext_email, ext_phone, loan_amount

Table:

<u>art_id</u> , art_type, items, quality, description, ext_id .
<u>ext_id</u> , ext_address, ext_name, ext_email, ext_phone, loan_amount

EXTRENAL MUSEUMS---*---TEMPORARY CONTRIBUTION---1--- EXHIBITIONS

UNF:

(ext_id, ext_address, ext_name, ext_email, ext_phone,
loan_amount, exh-id, exh_name, exh_type, display_cost, date, time,
ticket_price)

1NF:

Phone is a multivalued attribute.

(ext_id, ext_address, ext_name, ext_email, ext_phone,
loan_amount, exh-id, exh_name, exh_type, display cost, date, time,
ticket_price)

2NF:

1. ext_id, ext_address, ext_name, ext_e,mail, ext_phone, loan amount.
2. exh-id, exh_name, exh_type, display cost, date, time, ticket_price.

3NF:

1. ext_id, ext_address, ext_name, ext_e,mail, ext_phone, loan amount.
2. exh-id, exh_name, exh_type, display cost, date, time, ticket_price.

Table:

<u>ext_id</u> , ext_address, ext_name, ext_e,mail, ext_phone, loan amount, <u>exh_id</u> .
<u>exh-id</u> , exh_name, exh_type, display cost, date, time, ticket_price.

ART COLLECTIONS---*---SHOWCASE---*---ARTISTS

UNF:

(art_id, art_type, items, quality, description, ar_id, ar_name, ar_address, ar_phone, ar_email)

1NF:

Phone is a multivalued attribute.

(art_id, art_type, items, quality, description, ar_id, ar_name, ar_address, ar_phone, ar_email)

2NF:

1. art_id, art_type, items, quality, description.
2. ar_id, ar_name, ar_address, ar_phone, ar_email.

3NF:

1. art_id, art_type, items, quality, description.
2. ar_id, ar_name, ar_address, ar_phone, ar_email.

Table:

<u>art_id</u> , art_type, items, quality, description, <u>ar_id</u> .
<u>ar_id</u> , ar_name, ar_address, ar_phone, ar_email.

ARTISTS---*---PARTICIPATE---1--- EXHIBITIONS

UNF:

(ar_id, ar_name, ar_address, ar_phone, ar_email, exh-id, exh_type, exh_name, display_cost, date, time, ticket_price)

1NF:

Phone is a multivalued attribute.

(ar_id, ar_name, ar_address, ar_phone, ar_email, exh-id, exh_type, exh_name, display_cost, date, time, ticket_price)

2NF:

1. ar_id, ar_name, ar_address, ar_phone, ar_email.
2. exh-id, exh_type, exh_name, display_cost, date, time, ticket_price.

3NF:

1. ar_id, ar_name, ar_address, ar_phone, ar_email.
2. exh-id, exh_type, exh_name, display_cost, date, time, ticket_price.

Table:

<u>ar_id</u> , ar_name, ar_address, ar_phone, ar_email, <u>exh_id</u> .
<u>exh-id</u> , exh_type, exh_name, display_cost, date, time, ticket_price.

ART COLLECTIONS---*---MAINTENANCE---*--- MAINTENANCE EMPLOYEES

UNF:

(art_id, art_type, items, quality, description, m_id, m_name, m_address, m_phone, m_email, m_salary)

1NF:

Phone is a multivalued attribute.

(art_id, art_type, items, quality, description, m_id, m_name, m_address, m_phone, m_email, m_salary)

2NF:

1. art_id, art_type, items, quality, description.
2. m_id, m_name, m_address, m_phone, m_email, m_salary.

3NF:

1. art_id, art_type, items, quality, description.
2. m_id, m_name, m_address, m_phone, m_email, m_salary.

Table:

<u>art_id</u> , art_type, items, quality, description, <u>m_id</u> .
<u>m_id</u> , m_name, m_address, m_phone, m_email, m_salary

ART COLLECTIONS---*---PROTECTION---*---SECURITY EMPLOYEES

UNF:

(art_id, art_type, items, quality, description, s_id, s_name, s_address, s_phone, s_email, s_salary)

1NF:

Phone is a multivalued attribute.

(art_id, art_type, items, quality, description, s_id, s_name, s_address, s_phone, s_email, s_salary)

2NF:

1. art_id, art_type, items, quality, description.
2. s_id, s_name, s_address, s_phone, s_email, s_salary

3NF:

1. art_id, art_type, items, quality, description.
2. s_id, s_name, s_address, s_phone, s_email, s_salary

Table:

<u>art_id</u> , art_type, items, quality, description, <u>s_id</u> .
<u>s_id</u> , s_name, s_address, s_phone, s_email, s_salary

EXHIBITIONS---*---ORGANIZE---*---ORGANIZERS

UNF:

(exh-id, exh_name, exh_type, display cost, date, time, ticket price, o_id, o_name, o_address, o_phone, o_email, o_salary)

1NF:

Phone is a multivalued attribute.

(exh-id, exh_name, exh_type, display cost, date, time, ticket price, o_id, o_name, o_address, o_phone, o_email, o_salary)

2NF:

1. exh-id, exh_name, exh_type, display cost, date, time, ticket price.
2. o_id, o_name, o_address, o_phone, o_email, o_salary.

3NF:

1. exh-id, exh_name, exh_type, display cost, date, time, ticket price.
2. o_id, o_name, o_address, o_phone, o_email, o_salary.

Table:

<u>exh-id</u> , exh_name, exh_type, display cost, date, time, ticket price, <u>o_id</u> .
<u>o_id</u> , o_name, o_address, o_phone, o_email, o_salary.

EXHIBITIONS---*---VISIT---*---VISITORS

UNF:

(exh-id,exh_type, exh_name, display_cost, date, time,
ticket_price, v_id, v_name, payment_method , v_opinion)

1NF:

There is no multivalued value.

(exh-id,exh_type, exh_name, display_cost, date, time,
ticket_price, v_id, v_name, payment_method, v_opinion)

2NF:

1. exh-id,exh_type, exh_name, display_cost, date, time, ticket_price.
2. v_id, v_name, payment_method, v_opinion.

3NF:

1. exh-id,exh_type, exh_name, display_cost, date, time, ticket_price.
2. v_id, v_name, payment_method, v_opinion.

Table:

<u>exh-id</u> ,exh_type, exh_name, display_cost, date, time, ticket_price, <u>v_id</u> .
<u>v_id</u> , v_name, payment_method, v_opinion.

TABLE CREATION:

1. art_id, art_type, items, quality, description, **exh_id**.
2. exh_id, exh_name, exh_type, display_cost, exh_date, ticket_price.
3. art_id, art_type, items, quality, description, **ext_id**.
4. ext_id, ext_name, ext_address, ext_email, ext_phone, loan_amount
5. ext_id, ext_name, ext_address, ext_email, ext_phone, loan_amount, **exh-id**.
6. art_id, art_type, items, quality, description, **ar_id**.
7. ar_id, ar_name, ar_address, ar_phone, ar_email.
8. ar_id, ar_name, ar_address, ar_phone, ar_email, **exh_id**.
9. art_id, art_type, items, quality, description, **m_id**.
10. m_id, m_name, m_address, m_phone, m_email, m_salary
11. art_id, art_type, items, quality, description, **s_id**.
12. s_id, s_name, s_address, s_phone, s_email, s_salary
13. exh-id, exh_name, exh_type, display cost, date, time, ticket price, **o_id**.
14. o_id, o_name, o_address, o_phone, o_email, o_salary.
15. exh-id, exh_type, exh_name, display_cost, date, ticket_price, **v_id**.
16. v_id, v_name, v_opinion, payment_method.

TABLE CREATION:

1. CREATE TABLE EXHIBITIONS (EXH_ID NUMBER(6) CONSTRAINT PK_EXHIBITIONS PRIMARY KEY, EXH_NAME VARCHAR2(14), EXH_TYPE VARCHAR2(20), DISPLAY_COST NUMBER(10), EXH_DATE DATE, TICKET_PRICE NUMBER(5));

[illegible]

- ```
2. CREATE TABLE ARTCOLLECTIONS (ART_ID NUMBER(5) CONSTRAINT
PK_ARTCOLLECTIONS2 PRIMARY KEY, ART_TYPE VARCHAR2(10), ITEMS
NUMBER(5), QUALITY VARCHAR(20), DESCRIPTION VARCHAR2(20),
EXH_ID NUMBER(5) CONSTRAINT FK_EXHIBITIONS REFERENCES
EXHIBITIONS);
```

Results Explain Describe Saved SQL History

Object Type TABLE Object ART\_COLLECTIONS

| Table           | Column      | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|-----------------|-------------|-----------|--------|-----------|-------|-------------|----------|---------|---------|
| ART_COLLECTIONS | ART_ID      | Number    | -      | 6         | 0     | 1           | -        | -       | -       |
|                 | ART_TYPE    | Varchar2  | 20     | -         | -     | -           | ✓        | -       | -       |
|                 | ITEMS       | Number    | -      | 10        | 0     | -           | ✓        | -       | -       |
|                 | QUALITY     | Varchar2  | 20     | -         | -     | -           | ✓        | -       | -       |
|                 | DESCRIPTION | Varchar2  | 20     | -         | -     | -           | ✓        | -       | -       |
|                 | EXH_ID      | Number    | -      | 6         | 0     | -           | ✓        | -       | -       |

1 - 6

```
3. CREATE TABLE EXTRENAL_MUSEUMS (EXT_ID NUMBER(6) CONSTRAINT
PK_EXTRENAL_MUSEUMS PRIMARY KEY, EXT_NAME VARCHAR2(14) ,
EXT_ADDRESS VARCHAR2(20), EXH_TYPE VARCHAR2(20), EXT_EMAIL
VARCHAR2(20), EXT_PHONE NUMBER(11), LOAN_AMOUNT NUMBER(5));
```

Results Explain Describe Saved SQL History

Object Type TABLE Object EXTRENAL\_MUSEUMS

| Table            | Column      | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|------------------|-------------|-----------|--------|-----------|-------|-------------|----------|---------|---------|
| EXTRENAL_MUSEUMS | EXT_ID      | Number    | -      | 5         | 0     | 1           | -        | -       | -       |
|                  | EXT_NAME    | Varchar2  | 14     | -         | -     | -           | ✓        | -       | -       |
|                  | EXT_ADDRESS | Varchar2  | 20     | -         | -     | -           | ✓        | -       | -       |
|                  | EXHTYPE     | Varchar2  | 20     | -         | -     | -           | ✓        | -       | -       |
|                  | EXT_EMAIL   | Varchar2  | 20     | -         | -     | -           | ✓        | -       | -       |
|                  | EXT_PHONE   | Number    | -      | 11        | 0     | -           | ✓        | -       | -       |
|                  | LOAN_AMOUNT | Number    | -      | 5         | 0     | -           | ✓        | -       | -       |

1 - 7

```
4.. CREATE TABLE ARTISTS (AR_ID NUMBER(5) CONSTRAINT PK_ARTIST1
PRIMARY KEY, AR_NAME VARCHAR2(15), AR_ADDRESS VARCHAR2(25),
AR_EMAIL VARCHAR2(20), AR_PHONE NUMBER(15);
```

Results

Explain

Describe

Saved SQL

History

Object Type **TABLE** Object **ARTISTS**

| Table                   | Column                     | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|-------------------------|----------------------------|-----------|--------|-----------|-------|-------------|----------|---------|---------|
| <a href="#">ARTISTS</a> | <a href="#">AR_ID</a>      | Number    | -      | 5         | 0     | 1           | -        | -       | -       |
|                         | <a href="#">AR_NAME</a>    | Varchar2  | 15     | -         | -     | -           | ✓        | -       | -       |
|                         | <a href="#">AR_ADDRESS</a> | Varchar2  | 25     | -         | -     | -           | ✓        | -       | -       |
|                         | <a href="#">AR_EMAIL</a>   | Varchar2  | 20     | -         | -     | -           | ✓        | -       | -       |
|                         | <a href="#">AR_PHONE</a>   | Number    | -      | 15        | 0     | -           | ✓        | -       | -       |

1 - 5



```
5.. CREATE TABLE MAINTENANCE_EMPLOYEES (M_ID NUMBER(5)
CONSTRAINT PK_MAINTENANCE_EMPLOYEES PRIMARY KEY, M_NAME
VARCHAR2(15), M_ADDRESS VARCHAR2(25), M_EMAIL VARCHAR2(20),
M_PHONE NUMBER(15), M_SALARY NUMBER(10);
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **MAINTENANCE\_EMPLOYEES**

| Table                                 | Column                    | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---------------------------------------|---------------------------|-----------|--------|-----------|-------|-------------|----------|---------|---------|
| <a href="#">MAINTENANCE_EMPLOYEES</a> | <a href="#">M_ID</a>      | Number    | -      | 5         | 0     | 1           | -        | -       | -       |
|                                       | <a href="#">M_NAME</a>    | Varchar2  | 15     | -         | -     | -           | ✓        | -       | -       |
|                                       | <a href="#">M_ADDRESS</a> | Varchar2  | 25     | -         | -     | -           | ✓        | -       | -       |
|                                       | <a href="#">M_EMAIL</a>   | Varchar2  | 20     | -         | -     | -           | ✓        | -       | -       |
|                                       | <a href="#">M_PHONE</a>   | Number    | -      | 15        | 0     | -           | ✓        | -       | -       |
|                                       | <a href="#">M_SALARY</a>  | Number    | -      | 10        | 0     | -           | ✓        | -       | -       |

1 - 6

```
6.. CREATE TABLE SECURITY_EMPLOYEES (S_ID NUMBER(5) CONSTRAINT
PK_SECURITY_EMPLOYEES PRIMARY KEY, S_NAME VARCHAR2(15),
S_ADDRESS VARCHAR2(25), S_EMAIL VARCHAR2(20), S_PHONE
NUMBER(15), S_SALARY NUMBER(10);
```

|                         |                         |                          |                           |                         |
|-------------------------|-------------------------|--------------------------|---------------------------|-------------------------|
| <a href="#">Results</a> | <a href="#">Explain</a> | <a href="#">Describe</a> | <a href="#">Saved SQL</a> | <a href="#">History</a> |
|-------------------------|-------------------------|--------------------------|---------------------------|-------------------------|

---

Object Type **TABLE** Object **SECURITY\_EMPLOYEES**

| Table                     | Column           | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---------------------------|------------------|-----------|--------|-----------|-------|-------------|----------|---------|---------|
| <u>SECURITY_EMPLOYEES</u> | <u>S_ID</u>      | Number    | -      | 5         | 0     | 1           | -        | -       | -       |
|                           | <u>S_NAME</u>    | Varchar2  | 15     | -         | -     | -           | ✓        | -       | -       |
|                           | <u>S_ADDRESS</u> | Varchar2  | 25     | -         | -     | -           | ✓        | -       | -       |
|                           | <u>S_EMAIL</u>   | Varchar2  | 20     | -         | -     | -           | ✓        | -       | -       |
|                           | <u>S_PHONE</u>   | Number    | -      | 15        | 0     | -           | ✓        | -       | -       |
|                           | <u>S_SALARY</u>  | Number    | -      | 10        | 0     | -           | ✓        | -       | -       |

```
7.. CREATE TABLE ORGANIZERS(O_ID NUMBER(5) CONSTRAINT
PK_ORGANIZERS PRIMARY KEY, O_NAME VARCHAR2(15), O_ADDRESS
VARCHAR2(25), O_EMAIL VARCHAR2(20), O_PHONE NUMBER(15),
O_SALARY NUMBER(10);
```

Results Explain Describe Saved SQL History

Object Type TABLE Object ORGANIZERS

| Table      | Column    | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|------------|-----------|-----------|--------|-----------|-------|-------------|----------|---------|---------|
| ORGANIZERS | O_ID      | Number    | -      | 5         | 0     | 1           | -        | -       | -       |
|            | O_NAME    | Varchar2  | 15     | -         | -     | -           | ✓        | -       | -       |
|            | O_ADDRESS | Varchar2  | 25     | -         | -     | -           | ✓        | -       | -       |
|            | O_EMAIL   | Varchar2  | 20     | -         | -     | -           | ✓        | -       | -       |
|            | O_PHONE   | Number    | -      | 15        | 0     | -           | ✓        | -       | -       |
|            | O_SALARY  | Number    | -      | 10        | 0     | -           | ✓        | -       | -       |

1 - 6

```
8.. CREATE TABLE VISITORS (V_ID NUMBER(5) CONSTRAINT PK_VISITORS
PRIMARY KEY, V_NAME VARCHAR2(10), V_OPINION VARCHAR2(30),
PAYMENT_METHOD VARCHAR2(10);
```

|                         |                         |                          |                           |                         |
|-------------------------|-------------------------|--------------------------|---------------------------|-------------------------|
| <a href="#">Results</a> | <a href="#">Explain</a> | <a href="#">Describe</a> | <a href="#">Saved SQL</a> | <a href="#">History</a> |
|-------------------------|-------------------------|--------------------------|---------------------------|-------------------------|

|             |              |        |                 |
|-------------|--------------|--------|-----------------|
| Object Type | <b>TABLE</b> | Object | <b>VISITORS</b> |
|-------------|--------------|--------|-----------------|

| Table                    | Column                         | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|--------------------------|--------------------------------|-----------|--------|-----------|-------|-------------|----------|---------|---------|
| <a href="#">VISITORS</a> | <a href="#">V_ID</a>           | Number    | -      | 5         | 0     | 1           | -        | -       | -       |
|                          | <a href="#">V_NAME</a>         | Varchar2  | 20     | -         | -     | -           | ✓        | -       | -       |
|                          | <a href="#">V_OPINION</a>      | Varchar2  | 30     | -         | -     | -           | ✓        | -       | -       |
|                          | <a href="#">PAYMENT_METHOD</a> | Varchar2  | 10     | -         | -     | -           | ✓        | -       | -       |

1 - 4

## DATA INSERTION

### EXHIBITIONS:

INSERT INTO EXHIBITIONS VALUES (001,'EXHIBITION-1','COMMERCIAL',200000,TO\_DATE('05-01-2018','DD-MM-YYYY'),500,001,1);

INSERT INTO EXHIBITIONS VALUES (002,'EXHIBITION-2','COMMERCIAL',300000,TO\_DATE('06-01-2018','DD-MM-YYYY'),800,002,2);

INSERT INTO EXHIBITIONS VALUES (003,'EXHIBITION-3','COMMERCIAL',400000,TO\_DATE('07-01-2018','DD-MM-YYYY'),1000,003,3);

INSERT INTO EXHIBITIONS VALUES (004,'EXHIBITION-4','NON-COMMERCIAL',80000,TO\_DATE('08-01-2018','DD-MM-YYYY'),300,004,4);

INSERT INTO EXHIBITIONS VALUES (005,'EXHIBITION-5','NON-COMMERCIAL',50000,TO\_DATE('09-01-2018','DD-MM-YYYY'),200,005,5);

### ART COLLECTIONS:

INSERT INTO ART\_COLLECTIONS VALUES (001,'HUMAN',10,'HIGH RATED','MEN&WOMEN',001);

INSERT INTO ART\_COLLECTIONS VALUES (002,'ANIMALS',10,'HIGH RATED','WILD',002);

INSERT INTO ART\_COLLECTIONS VALUES (003,'NATURE',10,'HIGH RATED','PLANTS & SEA',003);

INSERT INTO ART\_COLLECTIONS VALUES (004,'SPORTS',15,'HIGH RATED','BASEBALL & RAGBI',004);

INSERT INTO ART\_COLLECTIONS VALUES (005,'TRAVEL',15,'HIGH RATED','HIKING',005);

### **EXTERNAL MUSEUMS:**

INSERT INTO EXTERNAL\_MUSEUMS VALUES (001,'MUSEUM-1','UTTARA','MUSEUM.UTTARA@GMAIL.COM',01712345678,10,1);

INSERT INTO EXTERNAL\_MUSEUMS VALUES (002,'MUSEUM-2','MIRPUR','MUSEUM.MIRPUR@GMAIL.COM',01712345677,10,2);

INSERT INTO EXTERNAL\_MUSEUMS VALUES (003,'MUSEUM-3','BASUNHARA','MUSEUM.BASUNDARA@GMAIL.COM',01712345676,10,3);

INSERT INTO EXTERNAL\_MUSEUMS VALUES (004,'MUSEUM-4','BANANI','MUSEUM.BANANI@GMAIL.COM',01712345675,10,4);

INSERT INTO EXTERNAL\_MUSEUMS VALUES (005,'MUSEUM-5','GULSHAN','MUSEUM.GULSHAN@GMAIL.COM',01712345674,10,5);

### **ARTISTS:**

INSERT INTO ARTISTS VALUES (001,'ASHIK','A\_BLOCK','ASHIK@GMAIL.COM',01712345671,1);

INSERT INTO ARTISTS VALUES (002,'JUNE','B\_BLOCK','JUNE@GMAIL.COM',01712345672,2);

INSERT INTO ARTISTS VALUES (003,'NOWRIN','C\_BLOCK','NOWRIN@GMAIL.COM',01712345673,3);

INSERT INTO ARTISTS VALUES (004,'GALIB','D\_BLOCK','GALIB@GMAIL.COM',01712345674,4);

INSERT INTO ARTISTS VALUES (005,'ABRAR','E\_BLOCK','ABRAR@GMAIL.COM',01712345675,5);

### **MAINTENANCE EMPLOYEES:**

```
INSERT INTO MAINTENANCE_EMPLOYEES VALUES (001,'HARRY',
'A_BLOCK','HARRY@GMAIL.COM',01712345676,5000);
```

```
INSERT INTO MAINTENANCE_EMPLOYEES VALUES (002,'POTTER',
'B_BLOCK','POTTER@GMAIL.COM',01712345677,4500);
```

```
INSERT INTO MAINTENANCE_EMPLOYEES VALUES (003,'JEN',
'C_BLOCK','JEN@GMAIL.COM',01712345678,4000);
```

```
INSERT INTO MAINTENANCE_EMPLOYEES VALUES(004,'POWEL',
'D_BLOCK','POWEL@GMAIL.COM',01712345679,3500);
```

```
INSERT INTO MAINTENANCE_EMPLOYEES VALUES(005,'CHRIS',
'E_BLOCK','CHRIS@GMAIL.COM',01712345681,3000);
```

### **SECURITY EMPLOYEES:**

```
INSERT INTO SECURITY_EMPLOYEES VALUES (001,'ROBART',
'A_BLOCK','ROBART@GMAIL.COM',01712345645,5000);
```

```
INSERT INTO SECURITY_EMPLOYEES VALUES (002,'CORNEL',
'B_BLOCK','CORNEL@GMAIL.COM',01712345677,4500);
```

```
INSERT INTO SECURITY_EMPLOYEES VALUES (3,'CHESTER',
'C_BLOCK','CHESTER@GMAIL.COM',1712345648,4000);
```

```
INSERT INTO SECURITY_EMPLOYEES VALUES (004,'MIKE',
'D_BLOCK','MIKE@GMAIL.COM',01712345676,3500);
```

```
INSERT INTO SECURITY_EMPLOYEES VALUES (005,'HAN',
'E_BLOCK','HAN@GMAIL.COM',01712345617,3000);
```

## **ORGANIZERS:**

```
INSERT INTO ORGANIZERS VALUES (1,'JENTHAN','A_BLOCK',
'JENTHAN@GMAIL.COM',01712345641,5000);
```

```
INSERT INTO ORGANIZERS VALUES (002,'MARK','B_BLOCK',
'MARK@GMAIL.COM',01712345672,4500);
```

```
INSERT INTO ORGANIZERS VALUES (003,'TONY','C_BLOCK',
'TONY@GMAIL.COM',01712345643,4000);
```

```
INSERT INTO ORGANIZERS VALUES (4,'SHERLOK','D_BLOCK',
'SHERLOK@GMAIL.COM',01712345674,3500);
```

```
INSERT INTO ORGANIZERS VALUES (005,'JAMES','E_BLOCK',
'JAMES@GMAIL.COM',01712345615,3000);
```

## **VISITORS:**

```
INSERT INTO VISITORS VALUES(001,'MAX','GOOD','CREDIT-CARD');
```

```
INSERT INTO VISITORS VALUES(002,'JONES','GOOD','CREDIT-
CARD');
```

```
INSERT INTO VISITORS VALUES(003,'ROCK','MEDIUM','CASH');
```

```
INSERT INTO VISITORS VALUES(004,'LOKI','AVERAGE','CASH');
```

```
INSERT INTO VISITORS VALUES(005,'ODIN','GOOD','CREDIT-
CARD');
```

## QUERY WRITING:

### SINGLE ROW FUNCTIONS:

1. Question: Display all data of art type animals.

Answer: select \* from art\_collections where lower (art\_type)= 'animals';

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
SELECT *
FROM ART_COLLECTIONS
WHERE LOWER(ART_TYPE)= 'animals';
```

**Results** Explain Describe Saved SQL History

| ART_ID | ART_TYPE | ITEMS | QUALITY    | DESCRIPTION | EXH_ID |
|--------|----------|-------|------------|-------------|--------|
| 2      | ANIMALS  | 10    | HIGH RATED | WILD        | 2      |

1 rows returned in 0.00 seconds [CSV Export](#)

2. Question: Display artists name, length of artists, replace gmail to yahoo of all artists.

Answer: select ar\_name, length (ar\_name), ar\_email, replace (ar\_email,'gmail','yahoo')from artists.

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
SELECT AR_NAME, LENGTH(AR_NAME), AR_EMAIL, REPLACE(AR_EMAIL, 'GMAIL', 'YAHOO')
FROM ARTISTS;
```

**Results** Explain Describe Saved SQL History

| AR_NAME | LENGTH(AR_NAME) | AR_EMAIL         | REPLACE(AR_EMAIL,'GMAIL','YAHOO') |
|---------|-----------------|------------------|-----------------------------------|
| ASHIK   | 5               | ASHIK@GMAIL.COM  | ASHIK@YAHOO.COM                   |
| JUNE    | 4               | JUNE@GMAIL.COM   | JUNE@YAHOO.COM                    |
| NOWRIN  | 6               | NOWRIN@GMAIL.COM | NOWRIN@YAHOO.COM                  |
| GALIB   | 5               | GALIB@GMAIL.COM  | GALIB@YAHOO.COM                   |
| ABRAR   | 5               | ABRAR@GMAIL.COM  | ABRAR@YAHOO.COM                   |

5 rows returned in 0.02 seconds [CSV Export](#)

3. Question: Display exhibitions name and remaining days to held the exhibitions.

Answer: select exh\_name,exh\_date, round((exh\_date-sysdate)) "days remaining" from exhibitions;

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
SELECT EXH_NAME,EXH_DATE, ROUND((EXH_DATE-SYSDATE)) "DAYS REMAINING"
FROM EXHIBITIONS;
```

**Results** Explain Describe Saved SQL History

| ART_ID | ART_TYPE | ITEMS | QUALITY    | DESCRIPTION | EXH_ID |
|--------|----------|-------|------------|-------------|--------|
| 2      | ANIMALS  | 10    | HIGH RATED | WILD        | 2      |

1 rows returned in 0.00 seconds [CSV Export](#)

## **GROUP FUNCTIONS:**

1. Question: Display average salary, minimum salary, maximum salary of all security employees.

Answer: select avg(s\_salary) average, min(s\_salary) minimum, max(s\_salary) maximum from security\_employees;



Home > SQL > SQL Commands

☒ Autocommit   Display 10 ▼

```
SELECT AVG(S_SALARY) AVERAGE, MIN(S_SALARY) MINIMUM, MAX(S_SALARY) MAXIMUM
FROM SECURITY_EMPLOYEES;
```

---

**Results**   Explain   Describe   Saved SQL   History

| AVERAGE | MINIMUM | MAXIMUM |
|---------|---------|---------|
| 4000    | 3000    | 5000    |

1 rows returned in 0.02 seconds   [CSV Export](#)

2. Question: Display total number of artists.

Answer: select count(ar\_name) "TOTAL ARTISTS" from artists.

Home > SQL > SQL Commands

☒ Autocommit   Display 10 ▼

```
SELECT COUNT(AR_NAME) "TOTAL ARTISTS"
FROM ARTISTS;
```

---

**Results**   Explain   Describe   Saved SQL   History

| TOTAL ARTISTS |
|---------------|
| 5             |

1 rows returned in 0.00 seconds   [CSV Export](#)

3. Question: Display opening date and ending date of exhibitions.

Answer: select min(exh\_date) "FIRST DAY", max(exh\_date) "LAST DAY"  
from exhibitions;

☒ Autocommit Display 10 ▼

```
SELECT MIN(EXH_DATE) "FIRST DAY", MAX(EXH_DATE) "LAST DAY"
FROM EXHIBITIONS;
```

**Results** Explain Describe Saved SQL History

| FIRST DAY | LAST DAY  |
|-----------|-----------|
| 05-JAN-19 | 09-JAN-19 |

1 rows returned in 0.00 seconds

[CSV Export](#)

## SUBQUERY:

1. Question: Display the employee who gets highest salary.

Answer: select m\_name, m\_salary from maintenance\_employees where  
m\_salary= (select max(m\_salary) from maintenance\_employees) ;

☒ Autocommit Display 10 ▼

```
SELECT M_NAME, M_SALARY
FROM MAINTENANCE_EMPLOYEES
WHERE M_SALARY= (SELECT MAX(M_SALARY)
 FROM MAINTENANCE_EMPLOYEES);
```

**Results** Explain Describe Saved SQL History

| M_NAME | M_SALARY |
|--------|----------|
| HARRY  | 5000     |

1 rows returned in 0.00 seconds

[CSV Export](#)

2. Question: Display exhibition, exhibition type and display cost of a exhibition which is being organized by organizer number 3.

Answer: select exh\_name, exh\_type, display\_cost from exhibitions1  
where o\_id = (select o\_id from organizers where o\_id=3);

Home > SQL > **SQL Commands**

---

☒ Autocommit   Display 10 ▼

```
SELECT EXH_NAME, EXH_TYPE, DISPLAY_COST
FROM EXHIBITIONS1
WHERE O_ID = (SELECT O_ID
 FROM ORGANIZERS
 WHERE O_ID=3);
```

---

**Results**   Explain   Describe   Saved SQL   History

---

| EXH_NAME     | EXH_TYPE   | DISPLAY_COST |
|--------------|------------|--------------|
| EXHIBITION-3 | COMMERCIAL | 400000       |

1 rows returned in 0.00 seconds   [CSV Export](#)

## JOINING:

1. Question: Display artists names, artists phone numbers, exhibitions date and exhibitions ticket price by using joining.

Answer: select a. ar\_name, a.ar\_phone, e.exh\_date, e.ticket\_price from  
artists1 a, exhibitions e where a.exh\_id = e.exh\_id ;

☒ Autocommit   Display 10 ▼

```
SELECT A.AR_NAME, A.AR_PHONE, E.EXH_DATE, E.TICKET_PRICE
FROM ARTISTS1 A, EXHIBITIONS E
WHERE A.EXH_ID = E.EXH_ID;
```

**Results**   Explain   Describe   Saved SQL   History

| AR_NAME | AR_PHONE   | EXH_DATE  | TICKET_PRICE |
|---------|------------|-----------|--------------|
| ASHIK   | 1712345671 | 05-JAN-19 | 500          |
| JUNE    | 1712345672 | 06-JAN-19 | 800          |
| NOWRIN  | 1712345673 | 07-JAN-19 | 1000         |
| GALIB   | 1712345674 | 08-JAN-19 | 300          |
| ABRAR   | 1712345675 | 09-JAN-19 | 200          |

5 rows returned in 0.00 seconds

[CSV Export](#)

2. Question: Display visitors names and their opinions regarding exhibitions name and exhibitions type by using joining.

Answer: select v.v\_name, v.v\_opinion, e.exh\_name, e.exh\_type from visitors v, exhibitions2 e where v.v\_id = e.v\_id;

☒ Autocommit   Display 10 ▼

```
SELECT V.V_NAME, V.V_OPINION, E.EXH_NAME, E.EXH_TYPE
FROM VISITORS V, EXHIBITIONS2 E
WHERE V.V_ID = E.V_ID;
```

**Results**   Explain   Describe   Saved SQL   History

| V_NAME | V_OPINION | EXH_NAME     | EXH_TYPE       |
|--------|-----------|--------------|----------------|
| MAX    | GOOD      | EXHIBITION-1 | COMMERCIAL     |
| JONES  | GOOD      | EXHIBITION-2 | COMMERCIAL     |
| ROCK   | MEDIUM    | EXHIBITION-3 | COMMERCIAL     |
| LOKI   | AVERAGE   | EXHIBITION-4 | NON-COMMERCIAL |
| ODIN   | GOOD      | EXHIBITION-5 | NON-COMMERCIAL |

5 rows returned in 0.00 seconds

[CSV Export](#)

## **CONCLUSION**

After finished this project we have learned many things like how to make a database, making scenario, normalization, query and lots of this. We all have enjoyed. Special thanks to JUENA AHMED NOSHIN mam.