Lab - 8 DB_Project_Assignment_ERD To Relational Model, DDL

Name: Ronit Jain (202001081)

Nipun Shah (202001096)

Group No: 2

Section No: 6

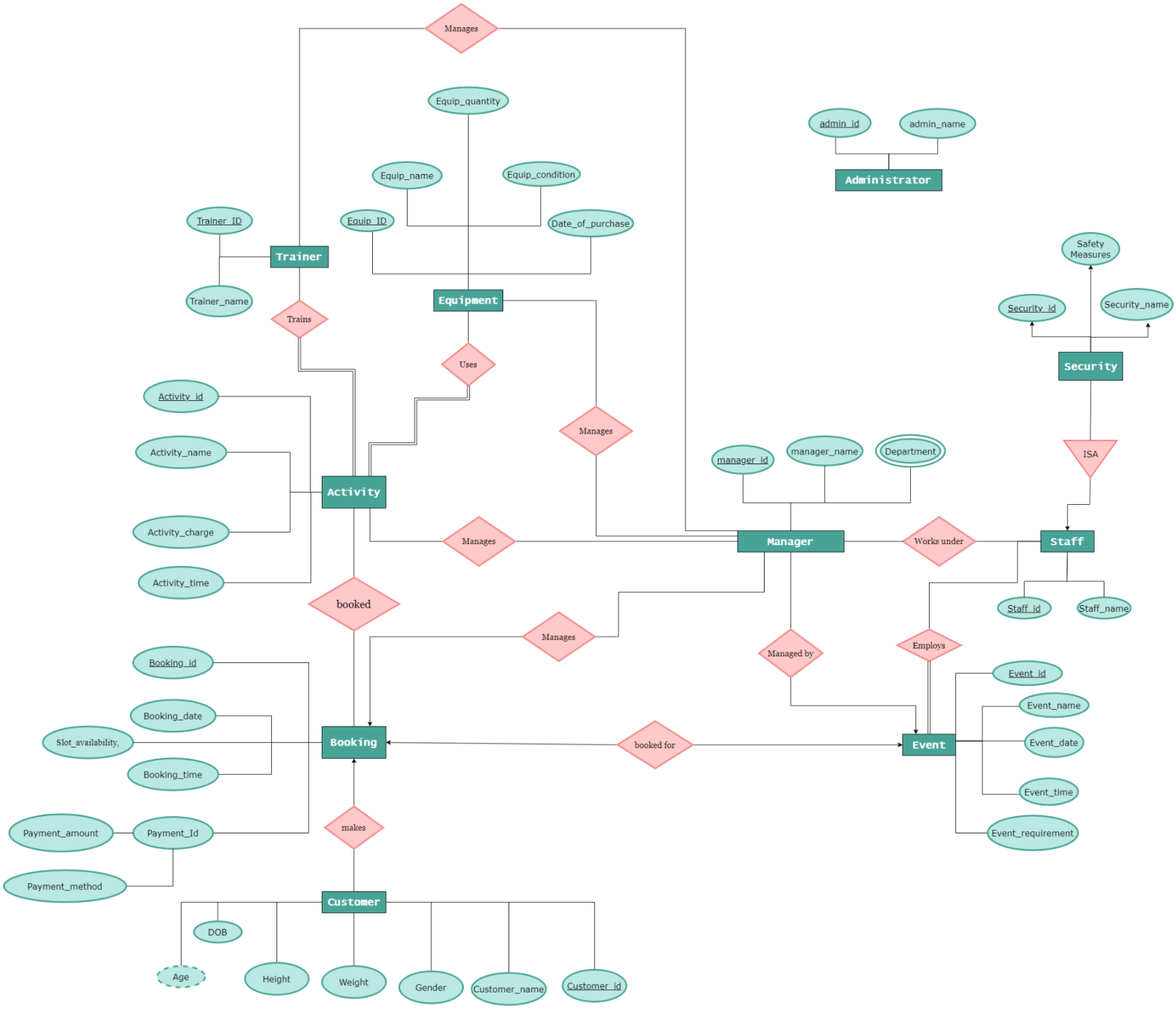
Team Id: 6.1

Case Study

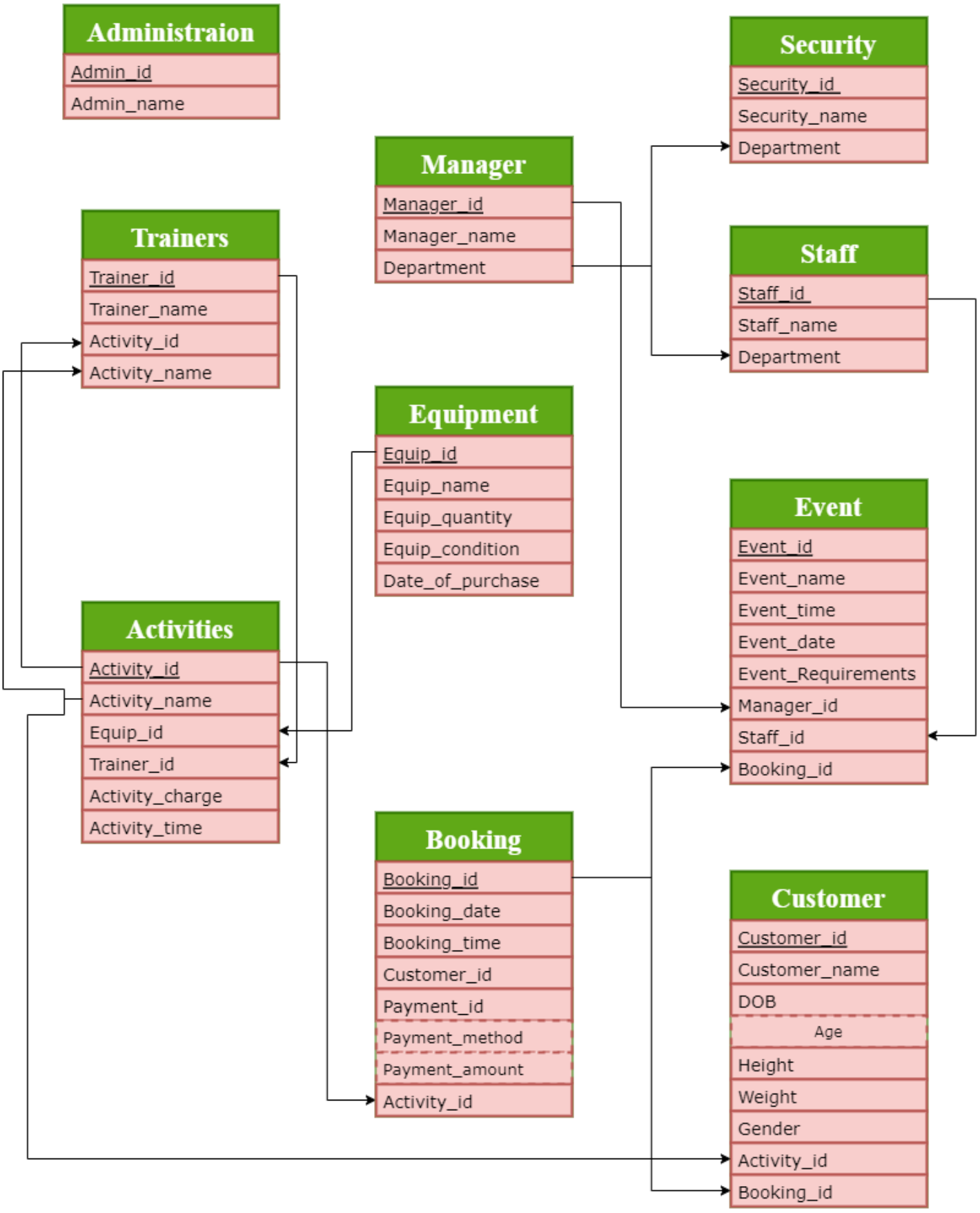
Beach Activity Management System:

All beach activity-related records should be created in this database. The beach-related activities like scuba diving, snorkeling, paragliding, flyboarding, windsurfing, and so on. Precautionary things like helmets, swimsuits, quality shoes, gloves, etc., for particular activities, must be appropriately managed for all customers. The other things related to taking pictures and videos are also included in this database.

Final ERD:


Manages
Equip_quantity
admin_id
admin_name
Administrator
Equip_name
Equip_condition
Trainer_ID
Equip_ID
Date_of_purchase
Trainer
Safety Measures
Trainer_name
Trains
Equipment
Security_id
Security_name
Security
Uses
Activity_id
Manages
ISA
manager_id
manager_name
Department
Activity_name
Activity
Activity_charge
Manages
Manager
Works under
Staff
Activity_time
booked
Staff_id
Staff_name
Manages
Managed by
Employs
Booking_id
Event_id
Booking_date
Event_name
Slot_availability
Booking
booked for
Event
Event_date
Booking_time
Event_time
makes
Payment_amount
Payment_Id
Event_requirement
Payment_method
Customer
DOB
Age
Height
Weight
Gender
Customer_name
Customer_id

Mapping E-R Model to Relational Model



Constraints:

In the above relational model we have 10 tables, each table has a primary key and some have foreign keys.

1. **Administration**

   Here, admin_id is the primary key which will uniquely identify the administrator.

2. **Manager**

   Here, manager_id is the primary key which will uniquely identify the manager.

3. **Activity**

   Here, activity_id is the primary key which will uniquely identify the activities. Here Equip_id and Trainer_id are the foreign keys from equipment and trainers table, which will help us to identify the equipment related to that activity and the trainers related to that specific activity.

4. **Equipment**

   Here, Equip_id is the primary key which will uniquely identify all the types of equipment present in the inventory.

5. **Booking**

   Here, Booking_id is the primary key which will uniquely identify all the bookings. In this table customer_id and activity_id will be foreign keys from the table customers and activities which will help us to identify the customer who has done the booking and the activity for which it has booked to participate.

6. **Customers**

   Here, customer_id is the primary key which will uniquely identify all the customers who have booked the activities or event. In This table we have activity_id and booking_id as foreign keys from the booking and activity table, which will help us to find the booking and activity of the customer.

7. **Trainer**

   Here, trainer_id is the primary key which will uniquely identify all the trainers. In this table we will have activity_id and activity_name from the activities table as the foreign key to identify the activity to which the trainer is related and will train that activity.

8. **Events**

   In this table, Event_id is the primary key which uniquely identifies all the events that are to be occurred. In this table we have booking_id from booking table, manager_id from manager table and staff_id from the staff table as foreign key so that we will about the manager who will be managing that event, booking_id will help us to see all the booking details and staff_id wll help us find the staff members who will be working for that particular event.

9. **Staff**

   In this table, staff_id is the primary key which will uniquely identify the staff member.

10. **Security**

    In this table, security_id is the primary key which will uniquely identify all the security staff. This table is also a child of the staff table.

DDL Script:

```
--- Activity


-- Table: ba_ms.Activity


-- DROP TABLE ba_ms."Activity";


CREATE TABLE ba_ms."Activity"
(
```

```
    "Activty_name" "char" NOT NULL,
    "Activity_id" character varying COLLATE pg_catalog."default" NOT
NULL,
    "Acticity_charge" numeric NOT NULL,
    "Activity_time" numeric NOT NULL,
    "Trainer_Id" character varying COLLATE pg_catalog."default" NOT
NULL,
    "Equipment_id" character varying COLLATE pg_catalog."default"
NOT NULL,
    CONSTRAINT "Activity_pkey" PRIMARY KEY ("Activity_id"),
    CONSTRAINT "Equipment_id" FOREIGN KEY ("Equipment_id")
        REFERENCES ba_ms."Equipment" ("Equipment_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "Trainer_Id" FOREIGN KEY ("Trainer_Id")
        REFERENCES ba_ms."Trainers" ("Trainer_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE ba_ms."Activity"
    OWNER to postgres;
-- Index: fki_Trainer_Id

-- DROP INDEX ba_ms."fki_Trainer_Id";
```

```
CREATE INDEX "fki_Trainer_Id"
    ON ba_ms."Activity" USING btree
    ("Trainer_Id" COLLATE pg_catalog."default" ASC NULLS LAST)
    TABLESPACE pg_default;




---Administration
-- Table: ba_ms.Administration

-- DROP TABLE ba_ms."Administration";

CREATE TABLE ba_ms."Administration"
(
    "Admin_id" character varying COLLATE pg_catalog."default" NOT
NULL,
    "Admin_name" "char" NOT NULL,
    CONSTRAINT "Administration_pkey" PRIMARY KEY ("Admin_id")
)

TABLESPACE pg_default;

ALTER TABLE ba_ms."Administration"
    OWNER to postgres;
```

```
---Booking


-- Table: ba_ms.Booking


-- DROP TABLE ba_ms."Booking";


CREATE TABLE ba_ms."Booking"
(
    "Booking_id" character varying COLLATE pg_catalog."default" NOT
NULL,
    "Booking_date" date NOT NULL,
    "Booking_time" time without time zone NOT NULL,
    "Payment_id" character varying COLLATE pg_catalog."default" NOT
NULL,
    "Payment_method" "char" NOT NULL,
    "Payment_amount" numeric NOT NULL,
    "Customer_id" character varying COLLATE pg_catalog."default" NOT
NULL,
    "Activity_id" character varying COLLATE pg_catalog."default" NOT
NULL,
    CONSTRAINT "Booking_pkey" PRIMARY KEY ("Booking_id"),
    CONSTRAINT "Activity_id" FOREIGN KEY ("Activity_id")
        REFERENCES ba_ms."Activity" ("Activity_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
```

```
    CONSTRAINT "Customer_id" FOREIGN KEY ("Customer_id")
        REFERENCES ba_ms."Customer" ("Customer_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE ba_ms."Booking"
    OWNER to postgres;



--Customer
-- Table: ba_ms.Customer

-- DROP TABLE ba_ms."Customer";

CREATE TABLE ba_ms."Customer"
(
    "Customer_id" character varying COLLATE pg_catalog."default" NOT
NULL,
    "Customer_name" "char" NOT NULL,
    "DOB" date[] NOT NULL,
    "Height" numeric[] NOT NULL,
    "Weight" numeric[] NOT NULL,
```

```
    "Gender" "char" NOT NULL,
    "Activity_id" character varying COLLATE pg_catalog."default" NOT
NULL,
    "Booking_id" character varying COLLATE pg_catalog."default" NOT
NULL,
    CONSTRAINT "Customer_pkey" PRIMARY KEY ("Customer_id"),
    CONSTRAINT "Activity_id" FOREIGN KEY ("Activity_id")
        REFERENCES ba_ms."Activity" ("Activity_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "Booking_id" FOREIGN KEY ("Booking_id")
        REFERENCES ba_ms."Booking" ("Booking_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE ba_ms."Customer"
    OWNER to postgres;




--Equipment
```

```
-- Table: ba_ms.Equipment

-- DROP TABLE ba_ms."Equipment";

CREATE TABLE ba_ms."Equipment"
(
    "Equipment_id" character varying COLLATE pg_catalog."default"
NOT NULL,
    "Equipment_name" "char" NOT NULL,
    "Equipment_quantity" numeric NOT NULL,
    "Equipment_condition" "char" NOT NULL,
    "Date_of_purchase" date NOT NULL,
    CONSTRAINT "Equipment_pkey" PRIMARY KEY ("Equipment_id")
)

TABLESPACE pg_default;

ALTER TABLE ba_ms."Equipment"
    OWNER to postgres;




--Event
-- Table: ba_ms.Event

-- DROP TABLE ba_ms."Event";
```

```
CREATE TABLE ba_ms."Event"

(

    "Event_id" character varying COLLATE pg_catalog."default" NOT
NULL,

    "Event_name" "char" NOT NULL,

    "Event_time" time without time zone NOT NULL,

    "Event_date" date NOT NULL,

    "Event_requirements" character varying COLLATE
pg_catalog."default" NOT NULL,

    "Manager_id" character varying COLLATE pg_catalog."default" NOT
NULL,

    "Staff_id" character varying COLLATE pg_catalog."default" NOT
NULL,

    "Booking_id" character varying COLLATE pg_catalog."default" NOT
NULL,

    CONSTRAINT "Event_pkey" PRIMARY KEY ("Event_id"),

    CONSTRAINT "Booking_id" FOREIGN KEY ("Booking_id")

        REFERENCES ba_ms."Booking" ("Booking_id") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID,

    CONSTRAINT "Manager_id" FOREIGN KEY ("Manager_id")

        REFERENCES ba_ms."Manager" ("Manager_id") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID,

    CONSTRAINT "Staff_id" FOREIGN KEY ("Staff_id")

        REFERENCES ba_ms."Staff" ("Staff_id") MATCH SIMPLE
```

```
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE ba_ms."Event"
    OWNER to postgres;




--Manageer

-- Table: ba_ms.Manager

-- DROP TABLE ba_ms."Manager";

CREATE TABLE ba_ms."Manager"
(
    "Manager_id" character varying COLLATE pg_catalog."default" NOT
NULL,
    "Manager_name" "char" NOT NULL,
    "Department" "char" NOT NULL,
    CONSTRAINT "Manager_id" PRIMARY KEY ("Manager_id")
)
```

```
TABLESPACE pg_default;

ALTER TABLE ba_ms."Manager"
    OWNER to postgres;




--Security
-- Table: ba_ms.Security

-- DROP TABLE ba_ms."Security";

CREATE TABLE ba_ms."Security"
(
    "Security_id" character varying COLLATE pg_catalog."default" NOT
NULL,
    "Security_name" "char" NOT NULL,
    "Department" "char" NOT NULL,
    CONSTRAINT "Security_pkey" PRIMARY KEY ("Security_id")
)

TABLESPACE pg_default;

ALTER TABLE ba_ms."Security"
    OWNER to postgres;
```

```
--Staff


-- Table: ba_ms.Staff


-- DROP TABLE ba_ms."Staff";


CREATE TABLE ba_ms."Staff"
(
    "Staff_id" character varying COLLATE pg_catalog."default" NOT
NULL,
    "Staff_name" "char" NOT NULL,
    "Department" "char" NOT NULL,
    CONSTRAINT "Staff_pkey" PRIMARY KEY ("Staff_id")
)


TABLESPACE pg_default;


ALTER TABLE ba_ms."Staff"
    OWNER to postgres;
```

```
--Trainers
-- Table: ba_ms.Trainers

-- DROP TABLE ba_ms."Trainers";

CREATE TABLE ba_ms."Trainers"
(
    "Trainer_id" character varying COLLATE pg_catalog."default" NOT
NULL,
    "Trainer_name" "char" NOT NULL,
    "Activity_id" character varying COLLATE pg_catalog."default" NOT
NULL,
    "Activity_name" "char" NOT NULL,
    CONSTRAINT "Trainers_pkey" PRIMARY KEY ("Trainer_id"),
    CONSTRAINT "Activity_id" FOREIGN KEY ("Activity_id")
        REFERENCES ba_ms."Activity" ("Activity_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE ba_ms."Trainers"
    OWNER to postgres;
```