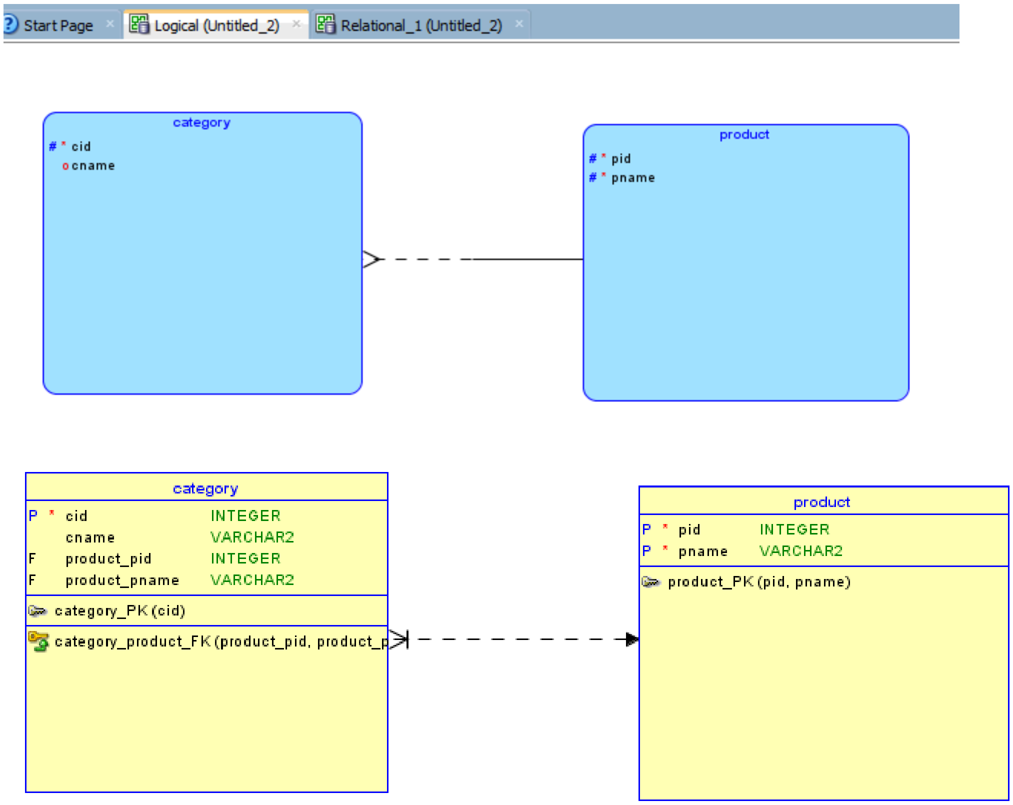
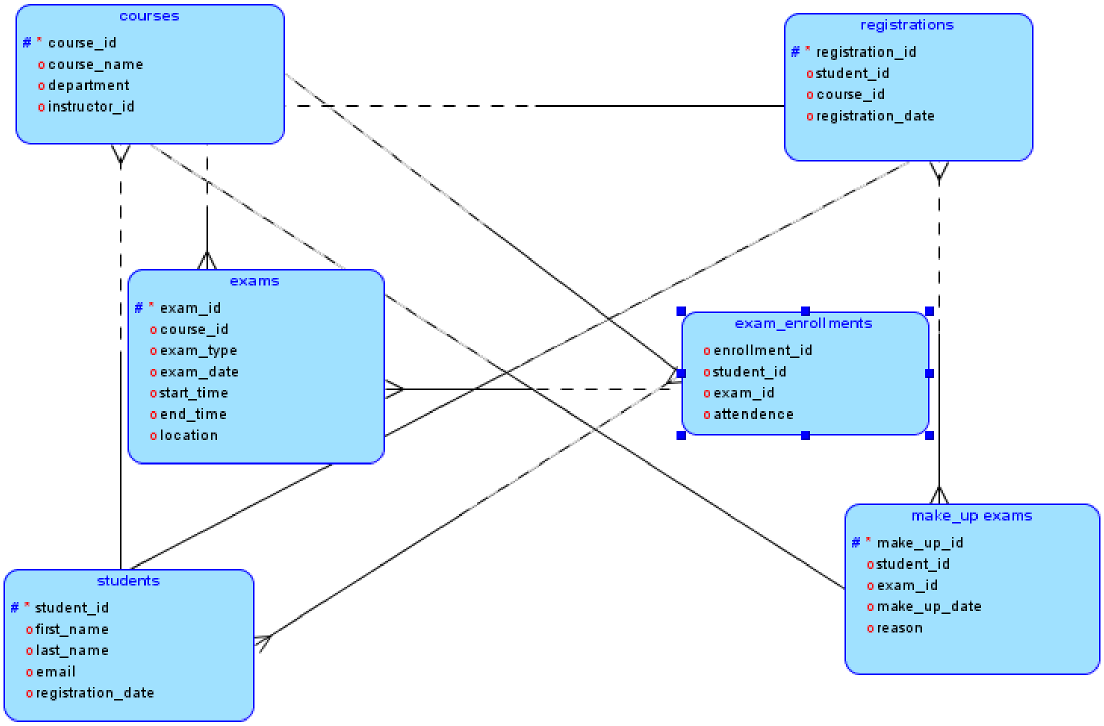
(CLASS TASKS)

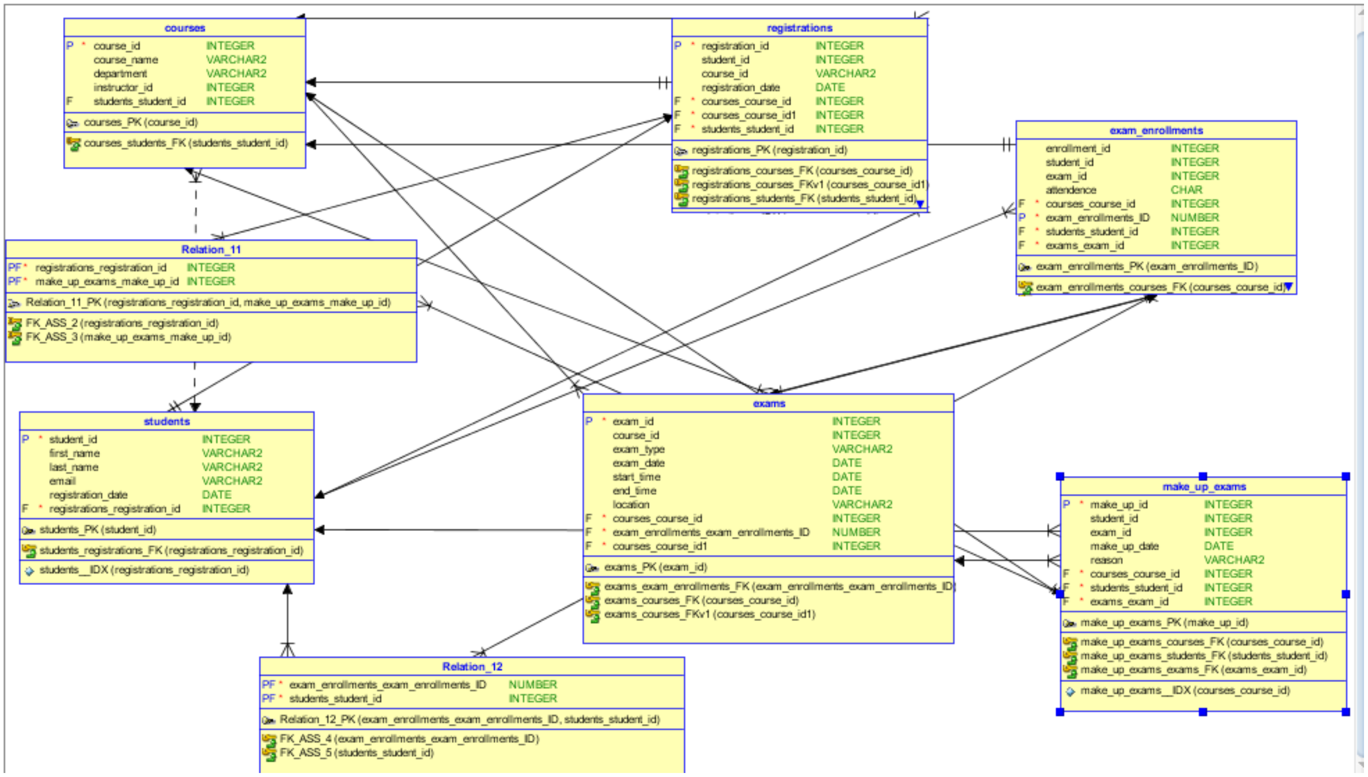
(practice task)



```
-- Generated by Oracle SQL Developer Data Modeler 4.1.1.888
--   at:        2024-10-02 13:56:32 PKT
--   site:      Oracle Database 11g
--   type:      Oracle Database 11g

CREATE
  TABLE category
  (
    cid   INTEGER NOT NULL ,
    cname VARCHAR2
    --  ERROR: VARCHAR2 size not specified
    ,
    product_pid   INTEGER ,
    product_pname VARCHAR2
    --  ERROR: VARCHAR2 size not specified
  ) ;
ALTER TABLE category ADD CONSTRAINT category_PK PRIMARY KEY ( cid ) ;


CREATE
  TABLE product
  (
    pid   INTEGER NOT NULL ,
    pname VARCHAR2
    --  ERROR: VARCHAR2 size not specified
    NOT NULL
  ) ;
ALTER TABLE product ADD CONSTRAINT product_PK PRIMARY KEY ( pid, pname ) ;


ALTER TABLE category ADD CONSTRAINT category_product_FK FOREIGN KEY (
product_pid, product_pname ) REFERENCES product ( pid, pname ) ;


-- Oracle SQL Developer Data Modeler Summary Report:
--
-- CREATE TABLE                  2
-- CREATE INDEX                  0
-- ALTER TABLE                  3
-- CREATE VIEW                  0
-- ALTER VIEW                  0
-- CREATE PACKAGE                  0
-- CREATE PACKAGE BODY                 0
-- CREATE PROCEDURE                  0
-- CREATE FUNCTION                 0
-- CREATE TRIGGER                  0
-- ALTER TRIGGER                  0
-- CREATE COLLECTION TYPE               0
-- CREATE STRUCTURED TYPE                0
-- CREATE STRUCTURED TYPE BODY              0
-- CREATE CLUSTER                  0
-- CREATE CONTEXT                  0
-- CREATE DATABASE                  0
-- CREATE DIMENSION                 0
-- CREATE DIRECTORY                 0
-- CREATE DISK GROUP                 0
-- CREATE ROLE                   0
-- CREATE ROLLBACK SEGMENT                0
-- CREATE SEQUENCE                  0
-- CREATE MATERIALIZED VIEW             0
-- CREATE SYNONYM                   0
-- CREATE TABLESPACE                 0
-- CREATE USER                   0
--
-- DROP TABLESPACE                  0
-- DROP DATABASE                  0
--
-- REDACTION POLICY                 0
--
-- ORDS DROP SCHEMA                  0
-- ORDS ENABLE SCHEMA                 0
-- ORDS ENABLE OBJECT                 0
--
-- ERRORS                     3
-- WARNINGS                    0
```

LAB MANUAL TASK (for 6 tables)




courses
P * course_id INTEGER
course_name VARCHAR2
department VARCHAR2
instructor_id INTEGER
F students_student_id INTEGER
courses_PK (course_id)
courses_students_FK (students_student_id)
registrations
P * registration_id INTEGER
student_id INTEGER
course_id VARCHAR2
registration_date DATE
F * courses_course_id INTEGER
F * courses_course_id1 INTEGER
F * students_student_id INTEGER
registrations_PK (registration_id)
registrations_courses_FK (courses_course_id)
registrations_courses_FKv1 (courses_course_id1)
registrations_students_FK (students_student_id)
exam_enrollments
enrollment_id INTEGER
student_id INTEGER
exam_id INTEGER
attendence CHAR
F * courses_course_id INTEGER
P * exam_enrollments_ID NUMBER
F * students_student_id INTEGER
F * exams_exam_id INTEGER
exam_enrollments_PK (exam_enrollments_ID)
exam_enrollments_courses_FK (courses_course_id)
Relation_11
PF * registrations_registration_id INTEGER
PF * make_up_exams_make_up_id INTEGER
Relation_11_PK (registrations_registration_id, make_up_exams_make_up_id)
FK_ASS_2 (registrations_registration_id)
FK_ASS_3 (make_up_exams_make_up_id)
students
P * student_id INTEGER
first_name VARCHAR2
last_name VARCHAR2
email VARCHAR2
registration_date DATE
F * registrations_registration_id INTEGER
students_PK (student_id)
students_registrations_FK (registrations_registration_id)
students__IDX (registrations_registration_id)
exams
P * exam_id INTEGER
course_id INTEGER
exam_type VARCHAR2
exam_date DATE
start_time DATE
end_time DATE
location VARCHAR2
F * courses_course_id INTEGER
F * exam_enrollments_exam_enrollments_ID NUMBER
F * courses_course_id1 INTEGER
exams_PK (exam_id)
exams_exam_enrollments_FK (exam_enrollments_exam_enrollments_ID)
exams_courses_FK (courses_course_id)
exams_courses_FKv1 (courses_course_id1)
make_up_exams
P * make_up_id INTEGER
student_id INTEGER
exam_id INTEGER
make_up_date DATE
reason VARCHAR2
F * courses_course_id INTEGER
F * students_student_id INTEGER
F * exams_exam_id INTEGER
make_up_exams_PK (make_up_id)
make_up_exams_courses_FK (courses_course_id)
make_up_exams_students_FK (students_student_id)
make_up_exams_exams_FK (exams_exam_id)
make_up_exams__IDX (courses_course_id)
Relation_12
PF * exam_enrollments_exam_enrollments_ID NUMBER
PF * students_student_id INTEGER
Relation_12_PK (exam_enrollments_exam_enrollments_ID, students_student_id)
FK_ASS_4 (exam_enrollments_exam_enrollments_ID)
FK_ASS_5 (students_student_id)

```
-- Generated by Oracle SQL Developer Data Modeler 4.1.1.888
--   at:        2024-10-02 15:38:37 PKT
--   site:      Oracle Database 11g
--   type:      Oracle Database 11g




CREATE
  TABLE Relation_11
  (
    registrations_registration_id INTEGER NOT NULL ,
    make_up_exams_make_up_id      INTEGER NOT NULL
  ) ;
ALTER TABLE Relation_11 ADD CONSTRAINT Relation_11_PK PRIMARY KEY (
registrations_registration_id, make_up_exams_make_up_id ) ;


CREATE
  TABLE Relation_12
  (
    --  ERROR: Column name length exceeds maximum allowed length(30)
    exam_enrollments_exam_enrollments_ID NUMBER NOT NULL ,
    students_student_id                  INTEGER NOT NULL
  ) ;
ALTER TABLE Relation_12 ADD CONSTRAINT Relation_12_PK PRIMARY KEY (
exam_enrollments_exam_enrollments_ID, students_student_id ) ;


CREATE
  TABLE courses
  (
    course_id   INTEGER NOT NULL ,
    course_name VARCHAR2
    --  ERROR: VARCHAR2 size not specified
    ,
    department VARCHAR2
    --  ERROR: VARCHAR2 size not specified
    ,
    instructor_id       INTEGER ,
    students_student_id INTEGER
  ) ;
ALTER TABLE courses ADD CONSTRAINT courses_PK PRIMARY KEY ( course_id ) ;


CREATE
  TABLE exam_enrollments
  (
    enrollment_id INTEGER ,
    student_id    INTEGER ,
    exam_id       INTEGER ,
    attendence    CHAR
    --  WARNING: CHAR size not specified
    ,
    courses_course_id   INTEGER NOT NULL ,
    exam_enrollments_ID NUMBER NOT NULL ,
    students_student_id INTEGER NOT NULL ,
    exams_exam_id       INTEGER NOT NULL
  ) ;
CREATE UNIQUE INDEX exam_enrollments__IDX ON exam_enrollments
  (
    courses_course_id ASC
  )
  ;
ALTER TABLE exam_enrollments ADD CONSTRAINT exam_enrollments_PK PRIMARY KEY (
exam_enrollments_ID ) ;


CREATE
  TABLE exams
  (
    exam_id   INTEGER NOT NULL ,
    course_id INTEGER ,
    exam_type VARCHAR2
    --  ERROR: VARCHAR2 size not specified
    ,
    exam_date  DATE ,
    start_time DATE ,
    end_time   DATE ,
    location   VARCHAR2
    --  ERROR: VARCHAR2 size not specified
    ,
    courses_course_id INTEGER NOT NULL ,
    --  ERROR: Column name length exceeds maximum allowed length(30)
    exam_enrollments_exam_enrollments_ID NUMBER NOT NULL ,
    courses_course_id1                   INTEGER NOT NULL
  ) ;
ALTER TABLE exams ADD CONSTRAINT exams_PK PRIMARY KEY ( exam_id ) ;


CREATE
  TABLE make_up_exams
  (
    make_up_id   INTEGER NOT NULL ,
    student_id   INTEGER ,
    exam_id      INTEGER ,
    make_up_date DATE ,
    reason       VARCHAR2
    --  ERROR: VARCHAR2 size not specified
    ,
    courses_course_id   INTEGER NOT NULL ,
    students_student_id INTEGER NOT NULL ,
    exams_exam_id       INTEGER NOT NULL
  ) ;
CREATE UNIQUE INDEX make_up_exams__IDX ON make_up_exams
  (
    courses_course_id ASC
  )
  ;
ALTER TABLE make_up_exams ADD CONSTRAINT make_up_exams_PK PRIMARY KEY (
make_up_id ) ;


CREATE
  TABLE registrations
  (
    registration_id INTEGER NOT NULL ,
    student_id      INTEGER ,
    course_id       VARCHAR2
    --  ERROR: VARCHAR2 size not specified
    ,
    registration_date   DATE ,
    courses_course_id   INTEGER NOT NULL ,
    courses_course_id1  INTEGER NOT NULL ,
    students_student_id INTEGER NOT NULL
  ) ;
CREATE UNIQUE INDEX registrations__IDX ON registrations
  (
    courses_course_id ASC
  )
  ;
ALTER TABLE registrations ADD CONSTRAINT registrations_PK PRIMARY KEY (
registration_id ) ;


CREATE
  TABLE students
  (
    student_id INTEGER NOT NULL ,
    first_name VARCHAR2
    --  ERROR: VARCHAR2 size not specified
    ,
    last_name VARCHAR2
    --  ERROR: VARCHAR2 size not specified
    ,
    email VARCHAR2
    --  ERROR: VARCHAR2 size not specified
    ,
    registration_date              DATE ,
    registrations_registration_id INTEGER NOT NULL
  ) ;
CREATE UNIQUE INDEX students__IDX ON students
  (
    registrations_registration_id ASC
  )
  ;
ALTER TABLE students ADD CONSTRAINT students_PK PRIMARY KEY ( student_id ) ;


ALTER TABLE Relation_11 ADD CONSTRAINT FK_ASS_2 FOREIGN KEY (
registrations_registration_id ) REFERENCES registrations ( registration_id ) ;
```

```
ALTER TABLE Relation_11 ADD CONSTRAINT FK_ASS_3 FOREIGN KEY (
make_up_exams_make_up_id ) REFERENCES make_up_exams ( make_up_id ) ;

ALTER TABLE Relation_12 ADD CONSTRAINT FK_ASS_4 FOREIGN KEY (
exam_enrollments_exam_enrollments_ID ) REFERENCES exam_enrollments (
exam_enrollments_ID ) ;

ALTER TABLE Relation_12 ADD CONSTRAINT FK_ASS_5 FOREIGN KEY (
students_student_id ) REFERENCES students ( student_id ) ;

ALTER TABLE courses ADD CONSTRAINT courses_students_FK FOREIGN KEY (
students_student_id ) REFERENCES students ( student_id ) ;

ALTER TABLE exam_enrollments ADD CONSTRAINT exam_enrollments_courses_FK FOREIGN
KEY ( courses_course_id ) REFERENCES courses ( course_id ) ;

ALTER TABLE exam_enrollments ADD CONSTRAINT exam_enrollments_exams_FK FOREIGN
KEY ( exams_exam_id ) REFERENCES exams ( exam_id ) ;

ALTER TABLE exam_enrollments ADD CONSTRAINT exam_enrollments_students_FK
FOREIGN KEY ( students_student_id ) REFERENCES students ( student_id ) ;

ALTER TABLE exams ADD CONSTRAINT exams_courses_FK FOREIGN KEY (
courses_course_id ) REFERENCES courses ( course_id ) ;

ALTER TABLE exams ADD CONSTRAINT exams_courses_FKv1 FOREIGN KEY (
courses_course_id1 ) REFERENCES courses ( course_id ) ;

ALTER TABLE exams ADD CONSTRAINT exams_exam_enrollments_FK FOREIGN KEY (
exam_enrollments_exam_enrollments_ID ) REFERENCES exam_enrollments (
exam_enrollments_ID ) ;

ALTER TABLE make_up_exams ADD CONSTRAINT make_up_exams_courses_FK FOREIGN KEY (
courses_course_id ) REFERENCES courses ( course_id ) ;

ALTER TABLE make_up_exams ADD CONSTRAINT make_up_exams_exams_FK FOREIGN KEY (
exams_exam_id ) REFERENCES exams ( exam_id ) ;

ALTER TABLE make_up_exams ADD CONSTRAINT make_up_exams_students_FK FOREIGN KEY
( students_student_id ) REFERENCES students ( student_id ) ;

ALTER TABLE registrations ADD CONSTRAINT registrations_courses_FK FOREIGN KEY (
courses_course_id ) REFERENCES courses ( course_id ) ;

ALTER TABLE registrations ADD CONSTRAINT registrations_courses_FKv1 FOREIGN KEY
( courses_course_id1 ) REFERENCES courses ( course_id ) ;

ALTER TABLE registrations ADD CONSTRAINT registrations_students_FK FOREIGN KEY
( students_student_id ) REFERENCES students ( student_id ) ;

ALTER TABLE students ADD CONSTRAINT students_registrations_FK FOREIGN KEY (
registrations_registration_id ) REFERENCES registrations ( registration_id ) ;

CREATE SEQUENCE exam_enrollments_exam_enrollme START WITH 1 NOCACHE ORDER ;
CREATE OR REPLACE TRIGGER exam_enrollments_exam_enrollme BEFORE
  INSERT
    ON exam_enrollments FOR EACH ROW WHEN
    (
      NEW.exam_enrollments_ID IS NULL
    )
    BEGIN :NEW.exam_enrollments_ID := exam_enrollments_exam_enrollme.NEXTVAL;
END;
/


-- Oracle SQL Developer Data Modeler Summary Report:
--
-- CREATE TABLE                             8
-- CREATE INDEX                             4
-- ALTER TABLE                             26
-- CREATE VIEW                              0
-- ALTER VIEW                               0
-- CREATE PACKAGE                           0
-- CREATE PACKAGE BODY                      0
-- CREATE PROCEDURE                         0
-- CREATE FUNCTION                          0
-- CREATE TRIGGER                           1
-- ALTER TRIGGER                            0
-- CREATE COLLECTION TYPE                   0
-- CREATE STRUCTURED TYPE                   0
-- CREATE STRUCTURED TYPE BODY              0
-- CREATE CLUSTER                           0
-- CREATE CONTEXT                           0
-- CREATE DATABASE                          0
-- CREATE DIMENSION                         0
-- CREATE DIRECTORY                         0
-- CREATE DISK GROUP                        0
-- CREATE ROLE                              0
-- CREATE ROLLBACK SEGMENT                  0
-- CREATE SEQUENCE                          1
-- CREATE MATERIALIZED VIEW                 0
-- CREATE SYNONYM                           0
-- CREATE TABLESPACE                        0
-- CREATE USER                              0
--
-- DROP TABLESPACE                          0
-- DROP DATABASE                            0
--
-- REDACTION POLICY                         0
--
-- ORDS DROP SCHEMA                         0
-- ORDS ENABLE SCHEMA                       0
-- ORDS ENABLE OBJECT                       0
--
-- ERRORS                                  11
-- WARNINGS                                 1
```

HOME TASKS (REMAINING SUBMISSION)

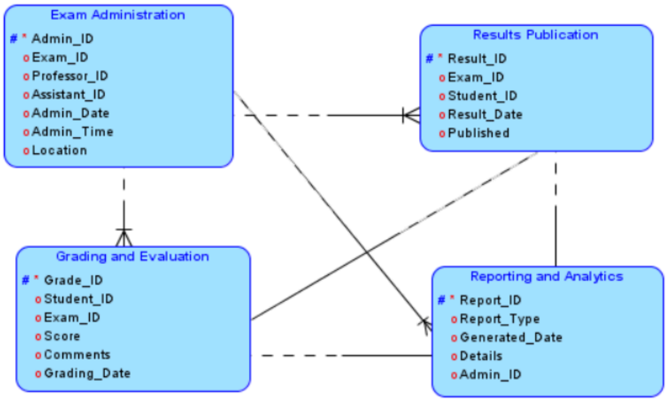
Lab Tasks-

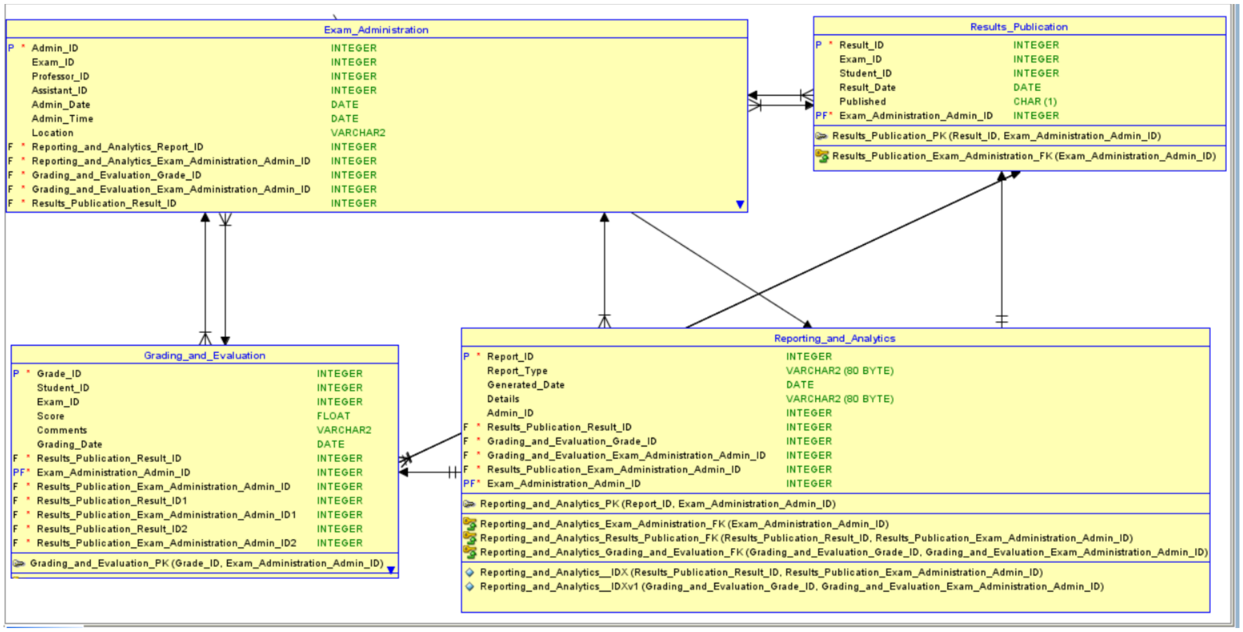
A prestigious university is revamping its student examination management system to modernize the process of conducting and managing examinations for a diverse student body. The new system is expected to handle all aspects of student assessments, including various types of examinations and evaluations. Here are the details,

5. **Exam Administration-** On the day of the exam, professors, teaching assistants, and invigilators play a role in administering the exams, ensuring adherence to university rules and regulations. The system should provide a means to record attendance and monitor the exam process.
6. **Grading and Evaluation-** After the exams are completed, professors and teaching assistants evaluate student responses. Some exams may involve multiple-choice questions, while others require subjective assessments. The system should facilitate grading and provide tools for professors to input scores and comments.
7. **Results Publication-** The university needs to publish exam results securely. Students should be able to access their results, and professors need access to individual and aggregate student performance data for further analysis.

10. **Reporting and Analytics-** The system should provide reporting and analytics features for the university administration to gain insights into student performance and course effectiveness.

To design the logical model, ER model, and generate DDL for this scenario, you will need to define entities, attributes, relationships, and business rules for each aspect of the scenario. The resulting database should efficiently store and manage information related to courses, exams, student registrations, exam scheduling, exam administration, grading, and exam results. Additionally, it should allow for scalability to accommodate the university's evolving examination needs.





Exam_Administration

P *	Admin_ID	INTEGER
	Exam_ID	INTEGER
	Professor_ID	INTEGER
	Assistant_ID	INTEGER
	Admin_Date	DATE
	Admin_Time	DATE
	Location	VARCHAR2
F *	Reporting_and_Analytics_Report_ID	INTEGER
F *	Reporting_and_Analytics_Exam_Administration_Admin_ID	INTEGER
F *	Grading_and_Evaluation_Grade_ID	INTEGER
F *	Grading_and_Evaluation_Exam_Administration_Admin_ID	INTEGER
F *	Results_Publication_Result_ID	INTEGER

Results_Publication

P *	Result_ID	INTEGER
	Exam_ID	INTEGER
	Student_ID	INTEGER
	Result_Date	DATE
	Published	CHAR (1)
PF*	Exam_Administration_Admin_ID	INTEGER

Results_Publication_PK (Result_ID, Exam_Administration_Admin_ID)

Results_Publication_Exam_Administration_FK (Exam_Administration_Admin_ID)

Grading_and_Evaluation

P *	Grade_ID	INTEGER
	Student_ID	INTEGER
	Exam_ID	INTEGER
	Score	FLOAT
	Comments	VARCHAR2
	Grading_Date	DATE
F *	Results_Publication_Result_ID	INTEGER
PF*	Exam_Administration_Admin_ID	INTEGER
F *	Results_Publication_Exam_Administration_Admin_ID	INTEGER
F *	Results_Publication_Result_ID1	INTEGER
F *	Results_Publication_Exam_Administration_Admin_ID1	INTEGER
F *	Results_Publication_Result_ID2	INTEGER
F *	Results_Publication_Exam_Administration_Admin_ID2	INTEGER

Grading_and_Evaluation_PK (Grade_ID, Exam_Administration_Admin_ID)

Reporting_and_Analytics

P *	Report_ID	INTEGER
	Report_Type	VARCHAR2 (80 BYTE)
	Generated_Date	DATE
	Details	VARCHAR2 (80 BYTE)
	Admin_ID	INTEGER
F *	Results_Publication_Result_ID	INTEGER
F *	Grading_and_Evaluation_Grade_ID	INTEGER
F *	Grading_and_Evaluation_Exam_Administration_Admin_ID	INTEGER
F *	Results_Publication_Exam_Administration_Admin_ID	INTEGER
PF*	Exam_Administration_Admin_ID	INTEGER

Reporting_and_Analytics_PK (Report_ID, Exam_Administration_Admin_ID)

Reporting_and_Analytics_Exam_Administration_FK (Exam_Administration_Admin_ID)
Reporting_and_Analytics_Results_Publication_FK (Results_Publication_Result_ID, Results_Publication_Exam_Administration_Admin_ID)
Reporting_and_Analytics_Grading_and_Evaluation_FK (Grading_and_Evaluation_Grade_ID, Grading_and_Evaluation_Exam_Administration_Admin_ID)

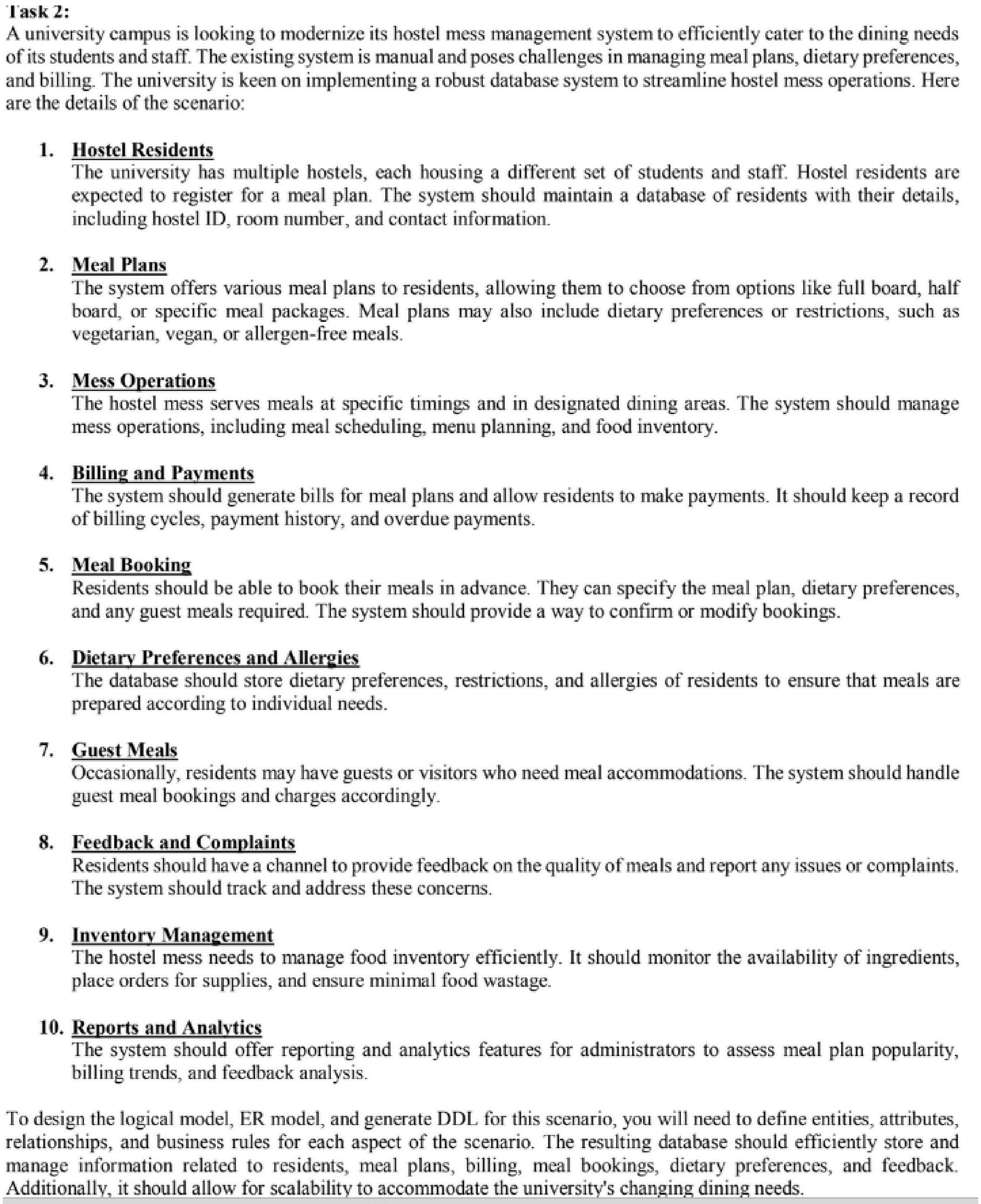
Reporting_and_Analytics__IDX (Results_Publication_Result_ID, Results_Publication_Exam_Administration_Admin_ID)
Reporting_and_Analytics__IDXv1 (Grading_and_Evaluation_Grade_ID, Grading_and_Evaluation_Exam_Administration_Admin_ID)

```
-- Generated by Oracle SQL Developer Data Modeler 4.1.1.888
--   at:        2024-10-06 19:23:36 PKT
--   site:      Oracle Database 11g
--   type:      Oracle Database 11g

CREATE
  TABLE Exam_Administration
  (
        Admin_ID            INTEGER NOT NULL ,
        Exam_ID             INTEGER ,
        Professor_ID INTEGER ,
        Assistant_ID INTEGER ,
        Admin_Date   DATE ,
        Admin_Time   DATE ,
        Location   VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Reporting_and_Analytics_Report_ID INTEGER NOT NULL ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Reporting_and_Analytics_Exam_Administration_Admin_ID INTEGER NOT NULL ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Grading_and_Evaluation_Grade_ID INTEGER NOT NULL ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Grading_and_Evaluation_Exam_Administration_Admin_ID INTEGER NOT NULL ,
        Results_Publication_Result_ID                  INTEGER NOT NULL ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Results_Publication_Exam_Administration_Admin_ID INTEGER NOT NULL
  ) ;
ALTER TABLE Exam_Administration ADD CONSTRAINT Exam_Administration_PK PRIMARY
KEY ( Admin_ID ) ;


CREATE
  TABLE Grading_and_Evaluation
  (
        Grade_ID   INTEGER NOT NULL ,
        Student_ID INTEGER ,
        Exam_ID  INTEGER ,
        Score FLOAT ,
        Comments VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Grading_Date                    DATE ,
        Results_Publication_Result_ID INTEGER NOT NULL ,
        Exam_Administration_Admin_ID  INTEGER NOT NULL ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Results_Publication_Exam_Administration_Admin_ID INTEGER NOT NULL ,
        Results_Publication_Result_ID1                 INTEGER NOT NULL ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Results_Publication_Exam_Administration_Admin_ID1 INTEGER NOT NULL ,
        Results_Publication_Result_ID2                 INTEGER NOT NULL ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Results_Publication_Exam_Administration_Admin_ID2 INTEGER NOT NULL
  ) ;
CREATE UNIQUE INDEX Grading_and_Evaluation__IDX ON Grading_and_Evaluation
  (
        Results_Publication_Result_ID ASC ,
        Results_Publication_Exam_Administration_Admin_ID ASC
  )
  ;
```

```
ALTER TABLE Grading_and_Evaluation ADD CONSTRAINT Grading_and_Evaluation_PK
PRIMARY KEY ( Grade_ID, Exam_Administration_Admin_ID ) ;


CREATE
  TABLE Reporting_and_Analytics
  (
          Report_ID                     INTEGER NOT NULL ,
          Report_Type                   VARCHAR2 (80 BYTE) ,
          Generated_Date                DATE ,
          Details             VARCHAR2 (80 BYTE) ,
          Admin_ID                      INTEGER ,
          Results_Publication_Result_ID INTEGER NOT NULL ,
          --  ERROR: Column name length exceeds maximum allowed length(30)
          Grading_and_Evaluation_Grade_ID INTEGER NOT NULL ,
          --  ERROR: Column name length exceeds maximum allowed length(30)
          Grading_and_Evaluation_Exam_Administration_Admin_ID INTEGER NOT NULL ,
          --  ERROR: Column name length exceeds maximum allowed length(30)
          Results_Publication_Exam_Administration_Admin_ID INTEGER NOT NULL ,
          Exam_Administration_Admin_ID                  INTEGER NOT NULL
  ) ;
CREATE UNIQUE INDEX Reporting_and_Analytics__IDX ON Reporting_and_Analytics
  (
          Results_Publication_Result_ID ASC ,
          Results_Publication_Exam_Administration_Admin_ID ASC
  )
  ;
CREATE UNIQUE INDEX Reporting_and_Analytics__IDXv1 ON Reporting_and_Analytics
  (
          Grading_and_Evaluation_Grade_ID ASC ,
          Grading_and_Evaluation_Exam_Administration_Admin_ID ASC
  )
  ;
ALTER TABLE Reporting_and_Analytics ADD CONSTRAINT Reporting_and_Analytics_PK
PRIMARY KEY ( Report_ID, Exam_Administration_Admin_ID ) ;


CREATE
  TABLE Results_Publication
  (
          Result_ID           INTEGER NOT NULL ,
          Exam_ID                       INTEGER ,
          Student_ID                    INTEGER ,
          Result_Date                   DATE ,
          Published           CHAR (1) ,
          Exam_Administration_Admin_ID INTEGER NOT NULL
  ) ;
ALTER TABLE Results_Publication ADD CONSTRAINT Results_Publication_PK PRIMARY
KEY ( Result_ID, Exam_Administration_Admin_ID ) ;


--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Exam_Administration ADD CONSTRAINT
Exam_Administration_Grading_and_Evaluation_FK FOREIGN KEY (
Grading_and_Evaluation_Grade_ID,
Grading_and_Evaluation_Exam_Administration_Admin_ID ) REFERENCES
Grading_and_Evaluation ( Grade_ID, Exam_Administration_Admin_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Exam_Administration ADD CONSTRAINT
Exam_Administration_Reporting_and_Analytics_FK FOREIGN KEY (
Reporting_and_Analytics_Report_ID,
Reporting_and_Analytics_Exam_Administration_Admin_ID ) REFERENCES
Reporting_and_Analytics ( Report_ID, Exam_Administration_Admin_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Exam_Administration ADD CONSTRAINT
Exam_Administration_Results_Publication_FK FOREIGN KEY (
Results_Publication_Result_ID, Results_Publication_Exam_Administration_Admin_ID
) REFERENCES Results_Publication ( Result_ID, Exam_Administration_Admin_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Grading_and_Evaluation ADD CONSTRAINT
Grading_and_Evaluation_Exam_Administration_FK FOREIGN KEY (
Exam_Administration_Admin_ID ) REFERENCES Exam_Administration ( Admin_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Grading_and_Evaluation ADD CONSTRAINT
Grading_and_Evaluation_Results_Publication_FK FOREIGN KEY (
Results_Publication_Result_ID, Results_Publication_Exam_Administration_Admin_ID
) REFERENCES Results_Publication ( Result_ID, Exam_Administration_Admin_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Grading_and_Evaluation ADD CONSTRAINT
Grading_and_Evaluation_Results_Publication_FKv1 FOREIGN KEY (
Results_Publication_Result_ID1,
Results_Publication_Exam_Administration_Admin_ID1 ) REFERENCES
Results_Publication ( Result_ID, Exam_Administration_Admin_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Grading_and_Evaluation ADD CONSTRAINT
Grading_and_Evaluation_Results_Publication_FKv2 FOREIGN KEY (
Results_Publication_Result_ID2,
Results_Publication_Exam_Administration_Admin_ID2 ) REFERENCES
Results_Publication ( Result_ID, Exam_Administration_Admin_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Reporting_and_Analytics ADD CONSTRAINT
Reporting_and_Analytics_Exam_Administration_FK FOREIGN KEY (
Exam_Administration_Admin_ID ) REFERENCES Exam_Administration ( Admin_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Reporting_and_Analytics ADD CONSTRAINT
Reporting_and_Analytics_Grading_and_Evaluation_FK FOREIGN KEY (
Grading_and_Evaluation_Grade_ID,
Grading_and_Evaluation_Exam_Administration_Admin_ID ) REFERENCES
Grading_and_Evaluation ( Grade_ID, Exam_Administration_Admin_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Reporting_and_Analytics ADD CONSTRAINT
Reporting_and_Analytics_Results_Publication_FK FOREIGN KEY (
Results_Publication_Result_ID, Results_Publication_Exam_Administration_Admin_ID
) REFERENCES Results_Publication ( Result_ID, Exam_Administration_Admin_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Results_Publication ADD CONSTRAINT
Results_Publication_Exam_Administration_FK FOREIGN KEY (
Exam_Administration_Admin_ID ) REFERENCES Exam_Administration ( Admin_ID ) ;


-- Oracle SQL Developer Data Modeler Summary Report:
--
```

```
-- CREATE TABLE                  4
-- CREATE INDEX                  3
-- ALTER TABLE                   15
-- CREATE VIEW                   0
-- ALTER VIEW                    0
-- CREATE PACKAGE                0
-- CREATE PACKAGE BODY                   0
-- CREATE PROCEDURE              0
-- CREATE FUNCTION               0
-- CREATE TRIGGER                0
-- ALTER TRIGGER                 0
-- CREATE COLLECTION TYPE                0
-- CREATE STRUCTURED TYPE                0
-- CREATE STRUCTURED TYPE BODY           0
-- CREATE CLUSTER                0
-- CREATE CONTEXT                0
-- CREATE DATABASE               0
-- CREATE DIMENSION              0
-- CREATE DIRECTORY              0
-- CREATE DISK GROUP             0
-- CREATE ROLE                   0
-- CREATE ROLLBACK SEGMENT               0
-- CREATE SEQUENCE               0
-- CREATE MATERIALIZED VIEW              0
-- CREATE SYNONYM                0
-- CREATE TABLESPACE             0
-- CREATE USER                   0
--
-- DROP TABLESPACE               0
-- DROP DATABASE                 0
--
-- REDACTION POLICY              0
--
-- ORDS DROP SCHEMA              0
-- ORDS ENABLE SCHEMA            0
-- ORDS ENABLE OBJECT            0
--
-- ERRORS                        24
-- WARNINGS                      0
```

TASK_2 (HOME_TASK)



Task 2:

A university campus is looking to modernize its hostel mess management system to efficiently cater to the dining needs of its students and staff. The existing system is manual and poses challenges in managing meal plans, dietary preferences, and billing. The university is keen on implementing a robust database system to streamline hostel mess operations. Here are the details of the scenario:

1. **Hostel Residents**
   The university has multiple hostels, each housing a different set of students and staff. Hostel residents are expected to register for a meal plan. The system should maintain a database of residents with their details, including hostel ID, room number, and contact information.

2. **Meal Plans**
   The system offers various meal plans to residents, allowing them to choose from options like full board, half board, or specific meal packages. Meal plans may also include dietary preferences or restrictions, such as vegetarian, vegan, or allergen-free meals.

3. **Mess Operations**
   The hostel mess serves meals at specific timings and in designated dining areas. The system should manage mess operations, including meal scheduling, menu planning, and food inventory.

4. **Billing and Payments**
   The system should generate bills for meal plans and allow residents to make payments. It should keep a record of billing cycles, payment history, and overdue payments.

5. **Meal Booking**
   Residents should be able to book their meals in advance. They can specify the meal plan, dietary preferences, and any guest meals required. The system should provide a way to confirm or modify bookings.

6. **Dietary Preferences and Allergies**
   The database should store dietary preferences, restrictions, and allergies of residents to ensure that meals are prepared according to individual needs.

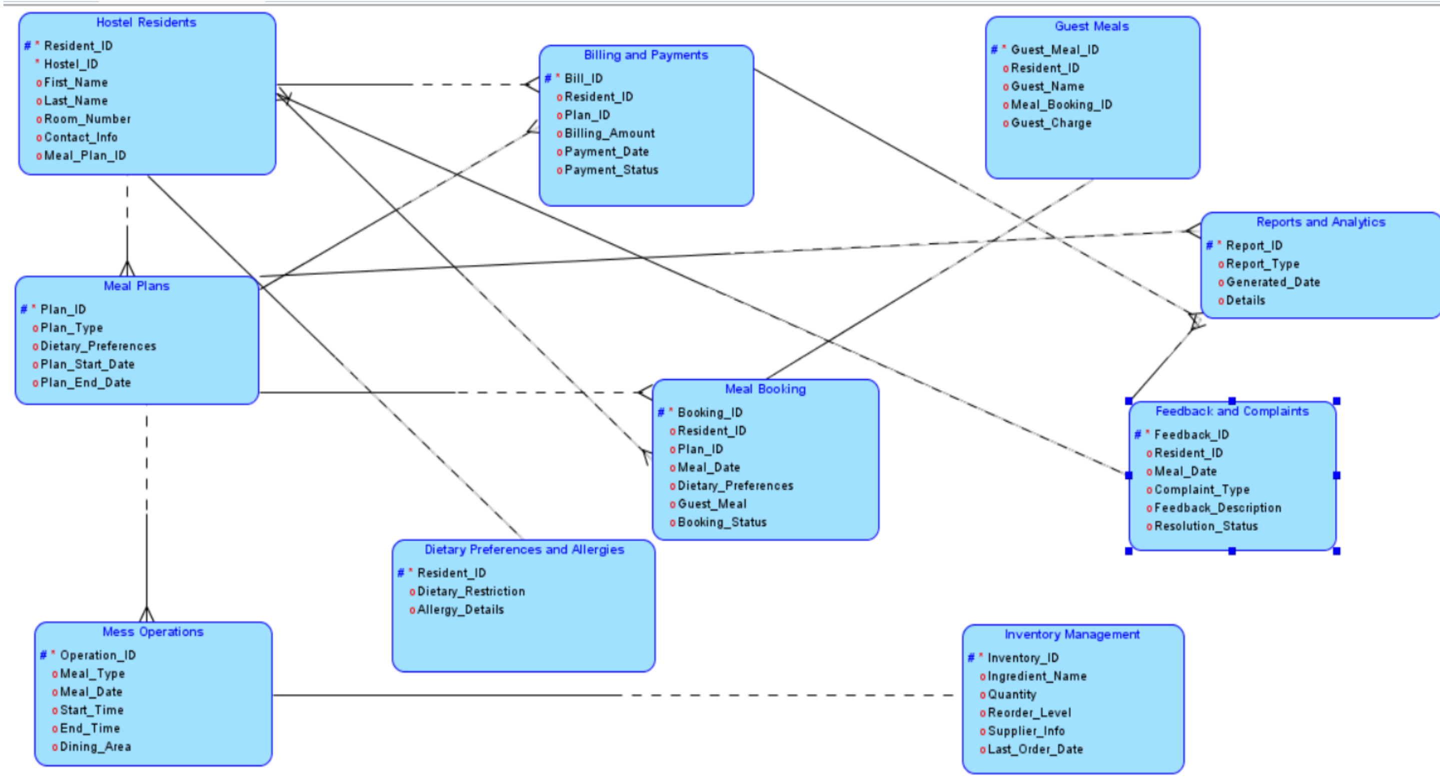
7. **Guest Meals**
   Occasionally, residents may have guests or visitors who need meal accommodations. The system should handle guest meal bookings and charges accordingly.

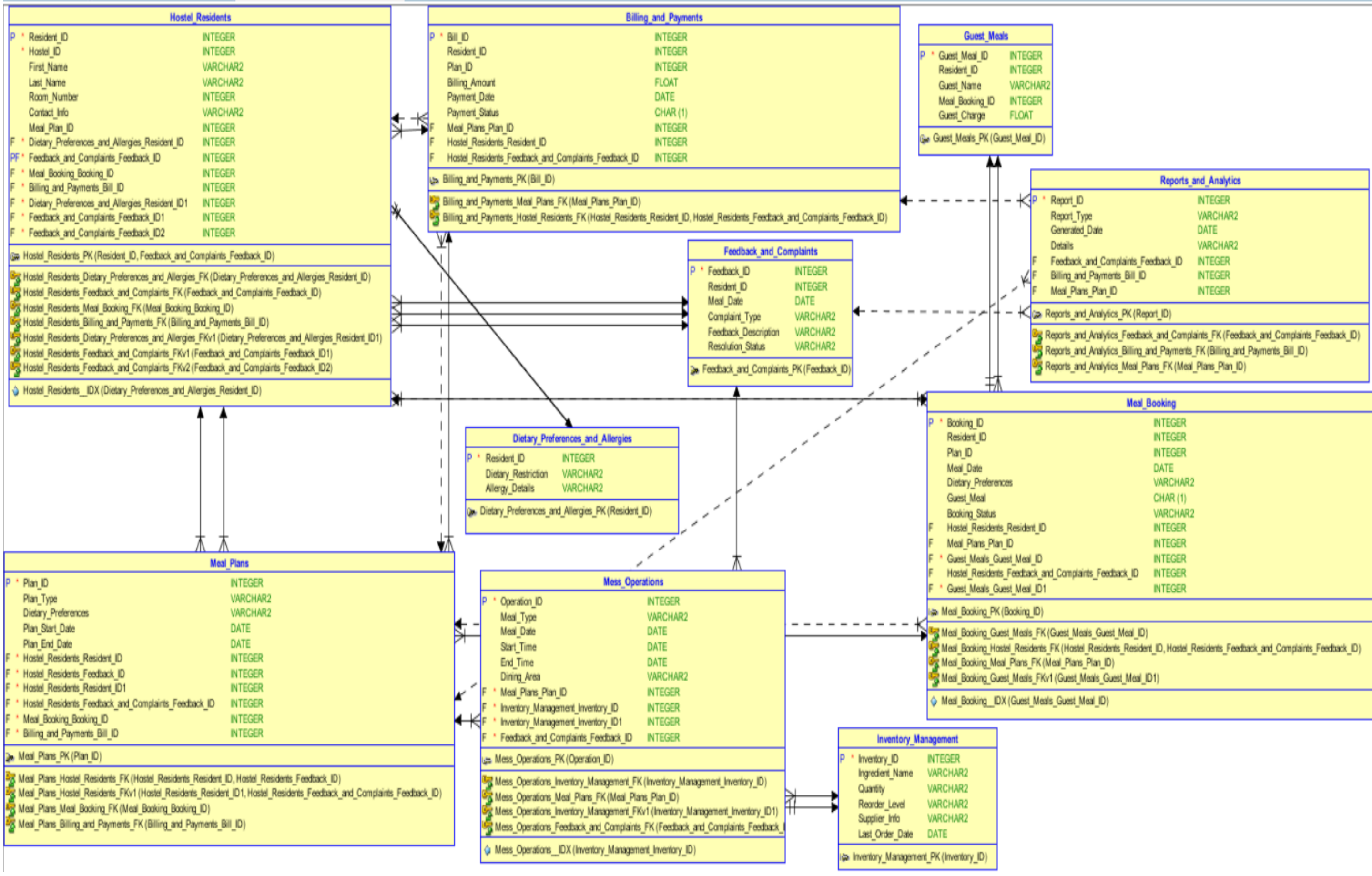
8. **Feedback and Complaints**
   Residents should have a channel to provide feedback on the quality of meals and report any issues or complaints. The system should track and address these concerns.

9. **Inventory Management**
   The hostel mess needs to manage food inventory efficiently. It should monitor the availability of ingredients, place orders for supplies, and ensure minimal food wastage.

10. **Reports and Analytics**
    The system should offer reporting and analytics features for administrators to assess meal plan popularity, billing trends, and feedback analysis.

To design the logical model, ER model, and generate DDL for this scenario, you will need to define entities, attributes, relationships, and business rules for each aspect of the scenario. The resulting database should efficiently store and manage information related to residents, meal plans, billing, meal bookings, dietary preferences, and feedback. Additionally, it should allow for scalability to accommodate the university's changing dining needs.


Hostel Residents
# * Resident_ID
* Hostel_ID
o First_Name
o Last_Name
o Room_Number
o Contact_Info
o Meal_Plan_ID
Billing and Payments
# * Bill_ID
o Resident_ID
o Plan_ID
o Billing_Amount
o Payment_Date
o Payment_Status
Guest Meals
# * Guest_Meal_ID
o Resident_ID
o Guest_Name
o Meal_Booking_ID
o Guest_Charge
Reports and Analytics
# * Report_ID
o Report_Type
o Generated_Date
o Details
Meal Plans
# * Plan_ID
o Plan_Type
o Dietary_Preferences
o Plan_Start_Date
o Plan_End_Date
Meal Booking
# * Booking_ID
o Resident_ID
o Plan_ID
o Meal_Date
o Dietary_Preferences
o Guest_Meal
o Booking_Status
Feedback and Complaints
# * Feedback_ID
o Resident_ID
o Meal_Date
o Complaint_Type
o Feedback_Description
o Resolution_Status
Dietary Preferences and Allergies
# * Resident_ID
o Dietary_Restriction
o Allergy_Details
Mess Operations
# * Operation_ID
o Meal_Type
o Meal_Date
o Start_Time
o End_Time
o Dining_Area
Inventory Management
# * Inventory_ID
o Ingredient_Name
o Quantity
o Reorder_Level
o Supplier_Info
o Last_Order_Date


Hostel_Residents
P * Resident_ID INTEGER
* Hostel_ID INTEGER
First_Name VARCHAR2
Last_Name VARCHAR2
Room_Number INTEGER
Contact_Info VARCHAR2
Meal_Plan_ID INTEGER
F * Dietary_Preferences_and_Allergies_Resident_ID INTEGER
PF * Feedback_and_Complaints_Feedback_ID INTEGER
F * Meal_Booking_Booking_ID INTEGER
F * Billing_and_Payments_Bill_ID INTEGER
F * Dietary_Preferences_and_Allergies_Resident_ID1 INTEGER
F * Feedback_and_Complaints_Feedback_ID1 INTEGER
F * Feedback_and_Complaints_Feedback_ID2 INTEGER
Hostel_Residents_PK (Resident_ID, Feedback_and_Complaints_Feedback_ID)
Hostel_Residents_Dietary_Preferences_and_Allergies_FK (Dietary_Preferences_and_Allergies_Resident_ID)
Hostel_Residents_Feedback_and_Complaints_FK (Feedback_and_Complaints_Feedback_ID)
Hostel_Residents_Meal_Booking_FK (Meal_Booking_Booking_ID)
Hostel_Residents_Billing_and_Payments_FK (Billing_and_Payments_Bill_ID)
Hostel_Residents_Dietary_Preferences_and_Allergies_FKv1 (Dietary_Preferences_and_Allergies_Resident_ID1)
Hostel_Residents_Feedback_and_Complaints_FKv1 (Feedback_and_Complaints_Feedback_ID1)
Hostel_Residents_Feedback_and_Complaints_FKv2 (Feedback_and_Complaints_Feedback_ID2)
Hostel_Residents__IDX (Dietary_Preferences_and_Allergies_Resident_ID)
Billing_and_Payments
P * Bill_ID INTEGER
Resident_ID INTEGER
Plan_ID INTEGER
Billing_Amount FLOAT
Payment_Date DATE
Payment_Status CHAR (1)
F Meal_Plans_Plan_ID INTEGER
F Hostel_Residents_Resident_ID INTEGER
F Hostel_Residents_Feedback_and_Complaints_Feedback_ID INTEGER
Billing_and_Payments_PK (Bill_ID)
Billing_and_Payments_Meal_Plans_FK (Meal_Plans_Plan_ID)
Billing_and_Payments_Hostel_Residents_FK (Hostel_Residents_Resident_ID, Hostel_Residents_Feedback_and_Complaints_Feedback_ID)
Guest_Meals
P * Guest_Meal_ID INTEGER
Resident_ID INTEGER
Guest_Name VARCHAR2
Meal_Booking_ID INTEGER
Guest_Charge FLOAT
Guest_Meals_PK (Guest_Meal_ID)
Reports_and_Analytics
P * Report_ID INTEGER
Report_Type VARCHAR2
Generated_Date DATE
Details VARCHAR2
F Feedback_and_Complaints_Feedback_ID INTEGER
F Billing_and_Payments_Bill_ID INTEGER
F Meal_Plans_Plan_ID INTEGER
Reports_and_Analytics_PK (Report_ID)
Reports_and_Analytics_Feedback_and_Complaints_FK (Feedback_and_Complaints_Feedback_ID)
Reports_and_Analytics_Billing_and_Payments_FK (Billing_and_Payments_Bill_ID)
Reports_and_Analytics_Meal_Plans_FK (Meal_Plans_Plan_ID)
Feedback_and_Complaints
P * Feedback_ID INTEGER
Resident_ID INTEGER
Meal_Date DATE
Complaint_Type VARCHAR2
Feedback_Description VARCHAR2
Resolution_Status VARCHAR2
Feedback_and_Complaints_PK (Feedback_ID)
Dietary_Preferences_and_Allergies
P * Resident_ID INTEGER
Dietary_Restriction VARCHAR2
Allergy_Details VARCHAR2
Dietary_Preferences_and_Allergies_PK (Resident_ID)
Meal_Booking
P * Booking_ID INTEGER
Resident_ID INTEGER
Plan_ID INTEGER
Meal_Date DATE
Dietary_Preferences VARCHAR2
Guest_Meal CHAR (1)
Booking_Status VARCHAR2
F Hostel_Residents_Resident_ID INTEGER
F Meal_Plans_Plan_ID INTEGER
F * Guest_Meals_Guest_Meal_ID INTEGER
F Hostel_Residents_Feedback_and_Complaints_Feedback_ID INTEGER
F * Guest_Meals_Guest_Meal_ID1 INTEGER
Meal_Booking_PK (Booking_ID)
Meal_Booking_Guest_Meals_FK (Guest_Meals_Guest_Meal_ID)
Meal_Booking_Hostel_Residents_FK (Hostel_Residents_Resident_ID, Hostel_Residents_Feedback_and_Complaints_Feedback_ID)
Meal_Booking_Meal_Plans_FK (Meal_Plans_Plan_ID)
Meal_Booking_Guest_Meals_FKv1 (Guest_Meals_Guest_Meal_ID1)
Meal_Booking__IDX (Guest_Meals_Guest_Meal_ID)
Meal_Plans
P * Plan_ID INTEGER
Plan_Type VARCHAR2
Dietary_Preferences VARCHAR2
Plan_Start_Date DATE
Plan_End_Date DATE
F * Hostel_Residents_Resident_ID INTEGER
F * Hostel_Residents_Feedback_ID INTEGER
F * Hostel_Residents_Resident_ID1 INTEGER
F * Hostel_Residents_Feedback_and_Complaints_Feedback_ID INTEGER
F * Meal_Booking_Booking_ID INTEGER
F * Billing_and_Payments_Bill_ID INTEGER
Meal_Plans_PK (Plan_ID)
Meal_Plans_Hostel_Residents_FK (Hostel_Residents_Resident_ID, Hostel_Residents_Feedback_ID)
Meal_Plans_Hostel_Residents_FKv1 (Hostel_Residents_Resident_ID1, Hostel_Residents_Feedback_and_Complaints_Feedback_ID)
Meal_Plans_Meal_Booking_FK (Meal_Booking_Booking_ID)
Meal_Plans_Billing_and_Payments_FK (Billing_and_Payments_Bill_ID)
Mess_Operations
P * Operation_ID INTEGER
Meal_Type VARCHAR2
Meal_Date DATE
Start_Time DATE
End_Time DATE
Dining_Area VARCHAR2
F * Meal_Plans_Plan_ID INTEGER
F * Inventory_Management_Inventory_ID INTEGER
F * Inventory_Management_Inventory_ID1 INTEGER
F * Feedback_and_Complaints_Feedback_ID INTEGER
Mess_Operations_PK (Operation_ID)
Mess_Operations_Inventory_Management_FK (Inventory_Management_Inventory_ID)
Mess_Operations_Meal_Plans_FK (Meal_Plans_Plan_ID)
Mess_Operations_Inventory_Management_FKv1 (Inventory_Management_Inventory_ID1)
Mess_Operations_Feedback_and_Complaints_FK (Feedback_and_Complaints_Feedback_
Mess_Operations__IDX (Inventory_Management_Inventory_ID)
Inventory_Management
P * Inventory_ID INTEGER
Ingredient_Name VARCHAR2
Quantity VARCHAR2
Reorder_Level VARCHAR2
Supplier_Info VARCHAR2
Last_Order_Date DATE
Inventory_Management_PK (Inventory_ID)

```
-- Generated by Oracle SQL Developer Data Modeler 4.1.1.888
--   at:        2024-10-06 22:07:17 PKT
--   site:      Oracle Database 11g
--   type:      Oracle Database 11g




CREATE TABLE Billing_and_Payments
  (
        Bill_ID       INTEGER NOT NULL ,
        Resident_ID INTEGER ,
        Plan_ID    INTEGER ,
        Billing_Amount FLOAT ,
        Payment_Date                  DATE ,
        Payment_Status                CHAR (1) ,
        Meal_Plans_Plan_ID            INTEGER ,
        Hostel_Residents_Resident_ID INTEGER ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Hostel_Residents_Feedback_and_Complaints_Feedback_ID INTEGER
  ) ;
ALTER TABLE Billing_and_Payments ADD CONSTRAINT Billing_and_Payments_PK PRIMARY KEY ( Bill_ID ) ;


--  ERROR: Table name length exceeds maximum allowed length(30)
CREATE TABLE Dietary_Preferences_and_Allergies
  (
        Resident_ID          INTEGER NOT NULL ,
        Dietary_Restriction VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Allergy_Details VARCHAR2
        --  ERROR: VARCHAR2 size not specified
  ) ;
--  ERROR: PK name length exceeds maximum allowed length(30)
ALTER TABLE Dietary_Preferences_and_Allergies ADD CONSTRAINT Dietary_Preferences_and_Allergies_PK PRIMARY KEY (
Resident_ID ) ;


CREATE TABLE Feedback_and_Complaints
  (
        Feedback_ID          INTEGER NOT NULL ,
        Resident_ID          INTEGER ,
        Meal_Date            DATE ,
        Complaint_Type VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Feedback_Description VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Resolution_Status VARCHAR2
        --  ERROR: VARCHAR2 size not specified
  ) ;
ALTER TABLE Feedback_and_Complaints ADD CONSTRAINT Feedback_and_Complaints_PK PRIMARY KEY ( Feedback_ID ) ;


CREATE TABLE Guest_Meals
  (
        Guest_Meal_ID INTEGER NOT NULL ,
        Resident_ID   INTEGER ,
        Guest_Name           VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Meal_Booking_ID INTEGER ,
        Guest_Charge FLOAT
  ) ;
ALTER TABLE Guest_Meals ADD CONSTRAINT Guest_Meals_PK PRIMARY KEY ( Guest_Meal_ID ) ;


CREATE TABLE Hostel_Residents
  (
        Resident_ID INTEGER NOT NULL ,
        Hostel_ID   INTEGER NOT NULL ,
        First_Name  VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Last_Name VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Room_Number  INTEGER ,
        Contact_Info VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Meal_Plan_ID INTEGER ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Dietary_Preferences_and_Allergies_Resident_ID INTEGER NOT NULL ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Feedback_and_Complaints_Feedback_ID INTEGER NOT NULL ,
        Meal_Booking_Booking_ID                    INTEGER NOT NULL ,
        Billing_and_Payments_Bill_ID               INTEGER NOT NULL ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Dietary_Preferences_and_Allergies_Resident_ID1 INTEGER NOT NULL ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Feedback_and_Complaints_Feedback_ID1 INTEGER NOT NULL ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Feedback_and_Complaints_Feedback_ID2 INTEGER NOT NULL
  ) ;
CREATE UNIQUE INDEX Hostel_Residents__IDX ON Hostel_Residents
  (
        Dietary_Preferences_and_Allergies_Resident_ID ASC
  )
  ;
ALTER TABLE Hostel_Residents ADD CONSTRAINT Hostel_Residents_PK PRIMARY KEY ( Resident_ID,
Feedback_and_Complaints_Feedback_ID ) ;


CREATE TABLE Inventory_Management
  (
        Inventory_ID           INTEGER NOT NULL ,
        Ingredient_Name VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Quantity VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Reorder_Level VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Supplier_Info VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Last_Order_Date DATE
  ) ;
ALTER TABLE Inventory_Management ADD CONSTRAINT Inventory_Management_PK PRIMARY KEY ( Inventory_ID ) ;


CREATE TABLE Meal_Booking
  (
        Booking_ID          INTEGER NOT NULL ,
        Resident_ID         INTEGER ,
        Plan_ID             INTEGER ,
        Meal_Date           DATE ,
        Dietary_Preferences VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Guest_Meal          CHAR (1) ,
        Booking_Status VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Hostel_Residents_Resident_ID INTEGER ,
        Meal_Plans_Plan_ID          INTEGER ,
        Guest_Meals_Guest_Meal_ID INTEGER NOT NULL ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Hostel_Residents_Feedback_and_Complaints_Feedback_ID INTEGER ,
        Guest_Meals_Guest_Meal_ID1                    INTEGER NOT NULL
  ) ;
CREATE UNIQUE INDEX Meal_Booking__IDX ON Meal_Booking
  (
        Guest_Meals_Guest_Meal_ID ASC
  )
  ;
ALTER TABLE Meal_Booking ADD CONSTRAINT Meal_Booking_PK PRIMARY KEY ( Booking_ID ) ;


CREATE TABLE Meal_Plans
  (
        Plan_ID   INTEGER NOT NULL ,
        Plan_Type VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Dietary_Preferences VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Plan_Start_Date             DATE ,
        Plan_End_Date               DATE ,
        Hostel_Residents_Resident_ID  INTEGER NOT NULL ,
        Hostel_Residents_Feedback_ID  INTEGER NOT NULL ,
        Hostel_Residents_Resident_ID1 INTEGER NOT NULL ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Hostel_Residents_Feedback_and_Complaints_Feedback_ID INTEGER NOT NULL ,
        Meal_Booking_Booking_ID                       INTEGER NOT NULL ,
        Billing_and_Payments_Bill_ID                  INTEGER NOT NULL
  ) ;
ALTER TABLE Meal_Plans ADD CONSTRAINT Meal_Plans_PK PRIMARY KEY ( Plan_ID ) ;


CREATE TABLE Mess_Operations
  (
        Operation_ID INTEGER NOT NULL ,
        Meal_Type           VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Meal_Date   DATE ,
        Start_Time  DATE ,
        End_Time          DATE ,
        Dining_Area VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Meal_Plans_Plan_ID INTEGER NOT NULL ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Inventory_Management_Inventory_ID INTEGER NOT NULL ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Inventory_Management_Inventory_ID1 INTEGER NOT NULL ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Feedback_and_Complaints_Feedback_ID INTEGER NOT NULL
  ) ;
CREATE UNIQUE INDEX Mess_Operations__IDX ON Mess_Operations
  (
        Inventory_Management_Inventory_ID ASC
  )
  ;
ALTER TABLE Mess_Operations ADD CONSTRAINT Mess_Operations_PK PRIMARY KEY ( Operation_ID ) ;


CREATE TABLE Reports_and_Analytics
  (
        Report_ID   INTEGER NOT NULL ,
        Report_Type VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        Generated_Date DATE ,
        Details     VARCHAR2
        --  ERROR: VARCHAR2 size not specified
        ,
        --  ERROR: Column name length exceeds maximum allowed length(30)
        Feedback_and_Complaints_Feedback_ID INTEGER ,
        Billing_and_Payments_Bill_ID          INTEGER ,
        Meal_Plans_Plan_ID           INTEGER
  ) ;
ALTER TABLE Reports_and_Analytics ADD CONSTRAINT Reports_and_Analytics_PK PRIMARY KEY ( Report_ID ) ;


--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Billing_and_Payments ADD CONSTRAINT Billing_and_Payments_Hostel_Residents_FK FOREIGN KEY (
Hostel_Residents_Resident_ID, Hostel_Residents_Feedback_and_Complaints_Feedback_ID ) REFERENCES Hostel_Residents (
Resident_ID, Feedback_and_Complaints_Feedback_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Billing_and_Payments ADD CONSTRAINT Billing_and_Payments_Meal_Plans_FK FOREIGN KEY (
Meal_Plans_Plan_ID ) REFERENCES Meal_Plans ( Plan_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Hostel_Residents ADD CONSTRAINT Hostel_Residents_Billing_and_Payments_FK FOREIGN KEY (
Billing_and_Payments_Bill_ID ) REFERENCES Billing_and_Payments ( Bill_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Hostel_Residents ADD CONSTRAINT Hostel_Residents_Dietary_Preferences_and_Allergies_FK FOREIGN KEY (
Dietary_Preferences_and_Allergies_Resident_ID ) REFERENCES Dietary_Preferences_and_Allergies ( Resident_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Hostel_Residents ADD CONSTRAINT Hostel_Residents_Dietary_Preferences_and_Allergies_FKv1 FOREIGN KEY
( Dietary_Preferences_and_Allergies_Resident_ID1 ) REFERENCES Dietary_Preferences_and_Allergies ( Resident_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
```

```
ALTER TABLE Hostel_Residents ADD CONSTRAINT Hostel_Residents_Feedback_and_Complaints_FK FOREIGN KEY (
Feedback_and_Complaints_Feedback_ID ) REFERENCES Feedback_and_Complaints ( Feedback_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Hostel_Residents ADD CONSTRAINT Hostel_Residents_Feedback_and_Complaints_FKv1 FOREIGN KEY (
Feedback_and_Complaints_Feedback_ID1 ) REFERENCES Feedback_and_Complaints ( Feedback_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Hostel_Residents ADD CONSTRAINT Hostel_Residents_Feedback_and_Complaints_FKv2 FOREIGN KEY (
Feedback_and_Complaints_Feedback_ID2 ) REFERENCES Feedback_and_Complaints ( Feedback_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Hostel_Residents ADD CONSTRAINT Hostel_Residents_Meal_Booking_FK FOREIGN KEY (
Meal_Booking_Booking_ID ) REFERENCES Meal_Booking ( Booking_ID ) ;

ALTER TABLE Meal_Booking ADD CONSTRAINT Meal_Booking_Guest_Meals_FK FOREIGN KEY (
Guest_Meals_Guest_Meal_ID ) REFERENCES Guest_Meals ( Guest_Meal_ID ) ;

ALTER TABLE Meal_Booking ADD CONSTRAINT Meal_Booking_Guest_Meals_FKv1 FOREIGN KEY (
Guest_Meals_Guest_Meal_ID1 ) REFERENCES Guest_Meals ( Guest_Meal_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Meal_Booking ADD CONSTRAINT Meal_Booking_Hostel_Residents_FK FOREIGN KEY (
Hostel_Residents_Resident_ID, Hostel_Residents_Feedback_and_Complaints_Feedback_ID ) REFERENCES Hostel_Residents (
Resident_ID, Feedback_and_Complaints_Feedback_ID ) ;

ALTER TABLE Meal_Booking ADD CONSTRAINT Meal_Booking_Meal_Plans_FK FOREIGN KEY ( Meal_Plans_Plan_ID )
REFERENCES Meal_Plans ( Plan_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Meal_Plans ADD CONSTRAINT Meal_Plans_Billing_and_Payments_FK FOREIGN KEY (
Billing_and_Payments_Bill_ID ) REFERENCES Billing_and_Payments ( Bill_ID ) ;

ALTER TABLE Meal_Plans ADD CONSTRAINT Meal_Plans_Hostel_Residents_FK FOREIGN KEY (
Hostel_Residents_Resident_ID, Hostel_Residents_Feedback_ID ) REFERENCES Hostel_Residents ( Resident_ID,
Feedback_and_Complaints_Feedback_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Meal_Plans ADD CONSTRAINT Meal_Plans_Hostel_Residents_FKv1 FOREIGN KEY (
Hostel_Residents_Resident_ID1, Hostel_Residents_Feedback_and_Complaints_Feedback_ID ) REFERENCES Hostel_Residents (
Resident_ID, Feedback_and_Complaints_Feedback_ID ) ;

ALTER TABLE Meal_Plans ADD CONSTRAINT Meal_Plans_Meal_Booking_FK FOREIGN KEY ( Meal_Booking_Booking_ID )
REFERENCES Meal_Booking ( Booking_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Mess_Operations ADD CONSTRAINT Mess_Operations_Feedback_and_Complaints_FK FOREIGN KEY (
Feedback_and_Complaints_Feedback_ID ) REFERENCES Feedback_and_Complaints ( Feedback_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Mess_Operations ADD CONSTRAINT Mess_Operations_Inventory_Management_FK FOREIGN KEY (
Inventory_Management_Inventory_ID ) REFERENCES Inventory_Management ( Inventory_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Mess_Operations ADD CONSTRAINT Mess_Operations_Inventory_Management_FKv1 FOREIGN KEY (
Inventory_Management_Inventory_ID1 ) REFERENCES Inventory_Management ( Inventory_ID ) ;

ALTER TABLE Mess_Operations ADD CONSTRAINT Mess_Operations_Meal_Plans_FK FOREIGN KEY ( Meal_Plans_Plan_ID )
REFERENCES Meal_Plans ( Plan_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
```

```
ALTER TABLE Reports_and_Analytics ADD CONSTRAINT Reports_and_Analytics_Billing_and_Payments_FK FOREIGN KEY (
Billing_and_Payments_Bill_ID ) REFERENCES Billing_and_Payments ( Bill_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Reports_and_Analytics ADD CONSTRAINT Reports_and_Analytics_Feedback_and_Complaints_FK FOREIGN KEY
( Feedback_and_Complaints_Feedback_ID ) REFERENCES Feedback_and_Complaints ( Feedback_ID ) ;

--  ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE Reports_and_Analytics ADD CONSTRAINT Reports_and_Analytics_Meal_Plans_FK FOREIGN KEY (
Meal_Plans_Plan_ID ) REFERENCES Meal_Plans ( Plan_ID ) ;


-- Oracle SQL Developer Data Modeler Summary Report:
--
-- CREATE TABLE                        10
-- CREATE INDEX                        3
-- ALTER TABLE                         34
-- CREATE VIEW                         0
-- ALTER VIEW                          0
-- CREATE PACKAGE                      0
-- CREATE PACKAGE BODY                          0
-- CREATE PROCEDURE                    0
-- CREATE FUNCTION                     0
-- CREATE TRIGGER                      0
-- ALTER TRIGGER                       0
-- CREATE COLLECTION TYPE                       0
-- CREATE STRUCTURED TYPE                       0
-- CREATE STRUCTURED TYPE BODY                  0
-- CREATE CLUSTER                      0
-- CREATE CONTEXT                      0
-- CREATE DATABASE                     0
-- CREATE DIMENSION                    0
-- CREATE DIRECTORY                    0
-- CREATE DISK GROUP                   0
-- CREATE ROLE                         0
-- CREATE ROLLBACK SEGMENT                      0
-- CREATE SEQUENCE                     0
-- CREATE MATERIALIZED VIEW                     0
-- CREATE SYNONYM                      0
-- CREATE TABLESPACE                   0
-- CREATE USER                         0
--
-- DROP TABLESPACE                     0
-- DROP DATABASE                       0
--
-- REDACTION POLICY                    0
--
-- ORDS DROP SCHEMA                    0
-- ORDS ENABLE SCHEMA                  0
-- ORDS ENABLE OBJECT                  0
--
-- ERRORS                              53
-- WARNINGS                            0
```