Database Foundations
6-3 : Defining Data Definition Language (DDL)
Practice Solutions

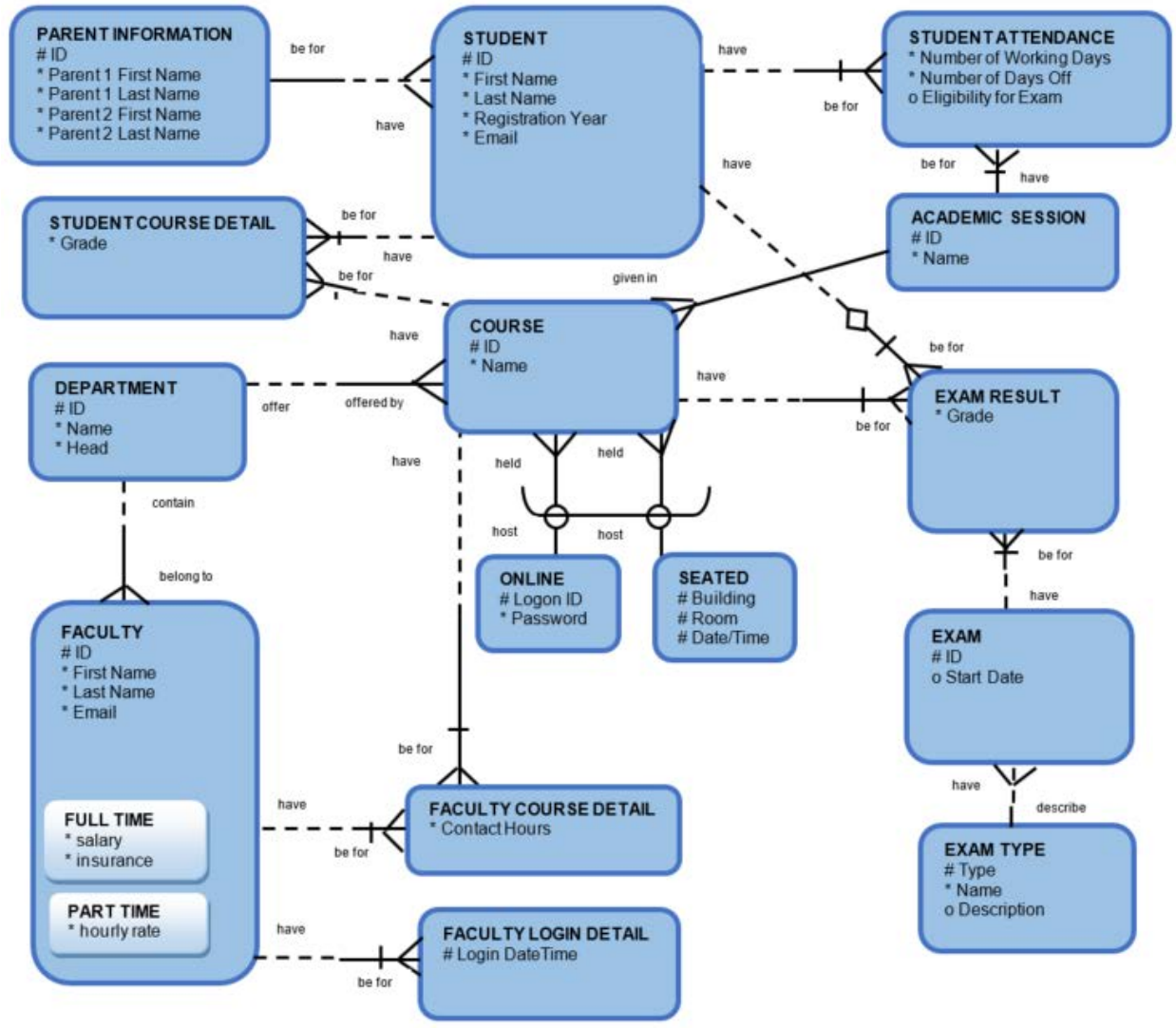
Exercise 1: Creating Tables Using Oracle Application Express

Overview

In this practice, you will create the tables for the Academic Database.

Assumptions

The following is the Entity Relationship Diagram (ERD) for the Academic Database where the tables will be created:



Tasks

1. Create the DDL Statements for creating the tables for the Academic Database listed above – include NOT NULL constraints where necessary. (Other constraints will be added later)

Solution:

```
CREATE TABLE AD_ACADEMIC_SESSIONS
  (
    ID            NUMBER,
    NAME          VARCHAR2(50) NOT NULL
  ) ;

CREATE TABLE AD_DEPARTMENTS
  (
    ID                  NUMBER ,
    NAME                VARCHAR2(50) NOT NULL,
    HEAD                VARCHAR2(50) NOT NULL
  ) ;

CREATE TABLE AD_PARENT_INFORMATION
  (
    ID                   NUMBER,
    PARENT1_FN          VARCHAR2(50) NOT NULL,
    PARENT1_LN          VARCHAR2(50) NOT NULL,
    PARENT2_FN          VARCHAR2(50) ,
    PARENT2_LN          VARCHAR2(50)
  ) ;

CREATE TABLE AD_STUDENTS
  (
     ID                 NUMBER ,
     FIRST_NAME         VARCHAR2(50) NOT NULL ,
     LAST_NAME          VARCHAR2(50) NOT NULL ,
     REG_YEAR           DATE NOT NULL,
     EMAIL              VARCHAR2(50) NOT NULL,
     PARENT_ID           NUMBER NOT NULL
  ) ;

CREATE TABLE AD_COURSES
  (
     ID                 NUMBER,
     NAME               VARCHAR2 (50) NOT NULL ,
     SESSION_ID         NUMBER  NOT NULL ,
     DEPT_ID            NUMBER ,
     LOGON_ID           VARCHAR2 (10),
     PASSWORD           VARCHAR2 (10) ,
     BUILDING           VARCHAR2 (50) ,
     ROOM               VARCHAR2 (20),
     DATE_TIME          VARCHAR2 (20)
  ) ;
```

```
CREATE TABLE AD_FACULTY
  (
     ID                  NUMBER ,
     FIRST_NAME          VARCHAR2(50) NOT NULL ,
     LAST_NAME           VARCHAR2(50) NOT NULL,
     EMAIL               VARCHAR2(50) NOT NULL,
     SALARY              NUMBER,
     INSURANCE           VARCHAR2(20),
     HOURLY_RATE         NUMBER,
     DEPT_ID             NUMBER
   ) ;

CREATE TABLE AD_EXAM_TYPES
  (
    TYPE                 VARCHAR2(50) ,
    NAME                 VARCHAR2(50) NOT NULL ,
    DESCRIPTION          VARCHAR2(50)0
  ) ;

CREATE TABLE AD_EXAMS
  (
    ID                   NUMBER ,
    START_DATE           DATE ,
    EXAM_TYPE            VARCHAR2(50) NOT NULL ,
    COURSE_ID            NUMBER  NOT NULL
   ) ;


CREATE TABLE AD_EXAM_RESULTS
  (
     STUDENT_ID          NUMBER ,
     COURSE_ID           NUMBER ,
     EXAM_ID             NUMBER ,
     EXAM_GRADE          NUMBER NOT NULL
  ) ;


CREATE TABLE AD_STUDENT_ATTENDANCE
  (
     STUDENT_ID          NUMBER ,
     SESSION_ID          NUMBER ,
     NUM_WORK_DAYS       NUMBER NOT NULL,
     NUM_DAYS_OFF        NUMBER NOT NULL,
     EXAM_ELIGIBILITY    VARCHAR2(20)
  ) ;
```

```
CREATE TABLE AD_STUDENT_COURSE_DETAILS
  (
     STUDENT_ID          NUMBER ,
     COURSE_ID           NUMBER ,
     GRADE               VARCHAR2(2)NOT NULL
  ) ;


CREATE TABLE AD_FACULTY_COURSE_DETAILS
  (
     FACULTY_ID          NUMBER ,
     COURSE_ID           NUMBER ,
     CONTACT_HRS         NUMBER NOT NULL
  ) ;


CREATE TABLE AD_FACULTY_LOGIN_DETAILS
  (
    FACULTY_ID           NUMBER ,
    LOGIN_DATE_TIME      TIMESTAMP NOT NULL
  ) ;
```

** Note : Completed script is available in Section 0 (Zip - Academic DB Script)

2. Run/execute these commands in **Oracle Application Express:**

Solution:

You can run/execute these commands in **Oracle Application Express** individually or by performing the following steps:

a. Login to APEX

b. In the navigation menu, click SQL Workshop/**SQL SCRIPTS**

c. Click UPLOAD and select the location of the .sql or text file that you stored your DDL code in. You can add a relevant name for you to identify the script

d. Once uploaded you will receive a report on the screen. Click RUN NOW

e. You will then have a list of your actions where you can click view results to see what has been processed.

f. You can view the database objects (tables) created in your schema by using Object Browser within SQL workshop.

Exercise 2: Altering the Tables

Overview

In this practice, you will:

- Alter the tables to set the constraints
- Specify a default value for a column
- Set a table to a read-only status

Assumptions

The primary and foreign key constraints are based on the ERD shown in the previous exercise and the unique constraints are based on the following :

The following fields should have unique values:

- Course Name in `AD_COURSES`
- Department Name in `AD_DEPARTMENTS`
- Student Email in `AD_STUDENTS`
- Faculty Email in `AD_FACULTY`
- Session Name in `AD_ACADEMIC_SESSIONS`

Tasks

1. Alter the tables in the Academic Database to define the primary key, foreign key and unique constraints.

Solution:

```
ALTER TABLE AD_ACADEMIC_SESSIONS ADD CONSTRAINT AD_ACADEMIC_SESSIONS_PK PRIMARY KEY (
SESSION_ID ) ;

ALTER TABLE AD_ACADEMIC_SESSIONS ADD CONSTRAINT AD_SESSIONS_NAME_UK UNIQUE( SESSION_NAME )
;

ALTER TABLE AD_COURSES ADD CONSTRAINT AD_COURSES_PK PRIMARY KEY (COURSE_ID) ;

ALTER TABLE AD_COURSES ADD CONSTRAINT AD_COURSES_NAME_UK UNIQUE(COURSE_NAME);

ALTER TABLE AD_COURSES ADD CONSTRAINT AD_COURSES_FK1 FOREIGN KEY ( SESSION_ID ) REFERENCES
AD_ACADEMIC_SESSIONS ( SESSION_ID ) ;

ALTER TABLE AD_COURSES ADD CONSTRAINT AD_COURSES_FK2 FOREIGN KEY ( DEPT_ID ) REFERENCES
AD_DEPARTMENTS ( DEPT_ID ) ;

ALTER TABLE AD_DEPARTMENTS ADD CONSTRAINT AD_DEPARTMENTS_PK PRIMARY KEY ( DEPT_ID ) ;

ALTER TABLE AD_DEPARTMENTS ADD CONSTRAINT AD_DEPT_NAME_UK UNIQUE( DEPT_NAME ) ;

ALTER TABLE AD_EXAMS ADD CONSTRAINT AD_EXAMS_PK PRIMARY KEY ( EXAM_ID ) ;

ALTER TABLE AD_EXAMS ADD CONSTRAINT AD_EXAMS_FK1 FOREIGN KEY ( EXAM_TYPE ) REFERENCES
AD_EXAM_TYPES ( EXAM_TYPE ) ;
```

```
ALTER TABLE AD_EXAMS ADD CONSTRAINT AD_EXAMS_FK2 FOREIGN KEY ( COURSE_ID ) REFERENCES
AD_COURSES ( COURSE_ID ) ;

ALTER TABLE AD_EXAM_TYPES ADD CONSTRAINT AD_EXAM_TYPES_PK PRIMARY KEY ( EXAM_TYPE ) ;

ALTER TABLE AD_EXAM_RESULTS ADD CONSTRAINT AD_EXAM_RESULTS_PK PRIMARY KEY ( STUDENT_ID,
COURSE_ID, EXAM_ID );

ALTER TABLE AD_EXAM_RESULTS ADD CONSTRAINT AD_EXAM_RESULTS_FK1 FOREIGN KEY ( STUDENT_ID )
REFERENCES AD_STUDENTS ( STUDENT_ID ) ;

ALTER TABLE AD_EXAM_RESULTS ADD CONSTRAINT AD_EXAM_RESULTS_FK2 FOREIGN KEY ( COURSE_ID )
REFERENCES AD_COURSES ( COURSE_ID ) ;

ALTER TABLE AD_EXAM_RESULTS ADD CONSTRAINT AD_EXAM_RESULT_FK3 FOREIGN KEY ( EXAM_ID )
REFERENCES AD_EXAMS ( EXAM_ID ) ;

ALTER TABLE AD_FACULTY ADD CONSTRAINT AD_FACULTY_PK PRIMARY KEY ( FACULTY_ID ) ;

ALTER TABLE AD_FACULTY ADD CONSTRAINT AD_FACULTY_FK FOREIGN KEY ( DEPT_ID ) REFERENCES
AD_DEPARTMENTS ( DEPT_ID ) ;

ALTER TABLE AD_FACULTY ADD CONSTRAINT AD_FACULTY_EMAIL_UK UNIQUE( FACULTY_EMAIL ) ;

ALTER TABLE AD_PARENT_INFORMATION ADD CONSTRAINT AD_PARENT_INFORMATION_PK PRIMARY KEY (
PARENT_ID ) ;

ALTER TABLE AD_STUDENTS ADD CONSTRAINT AD_STUDENTS_PK PRIMARY KEY ( STUDENT_ID ) ;

ALTER TABLE AD_STUDENTS ADD CONSTRAINT AD_STUDENTS_FK FOREIGN KEY ( PARENT_ID ) REFERENCES
AD_PARENT_INFORMATION ( PARENT_ID ) ;
ALTER TABLE AD_STUDENTS ADD CONSTRAINT AD_STUDENTS_EMAIL_UK UNIQUE(EMAIL );

ALTER TABLE AD_STUDENT_ATTENDANCE ADD CONSTRAINT AD_STUDENT_ATTENDANCE_PK PRIMARY KEY (
STUDENT_ID, SESSION_ID) ;

ALTER TABLE AD_STUDENT_ATTENDANCE ADD CONSTRAINT AD_STUDENT_ATTENDANCE_FK1 FOREIGN KEY (
STUDENT_ID ) REFERENCES AD_STUDENTS ( STUDENT_ID ) ;

ALTER TABLE AD_STUDENT_ATTENDANCE ADD CONSTRAINT AD_STUDENT_ATTENDANCE_FK2 FOREIGN KEY (
SESSION_ID ) REFERENCES AD_ACADEMIC_SESSIONS ( SESSION_ID ) ;

ALTER TABLE AD_STUDENT_COURSE_DETAILS ADD CONSTRAINT AD_STUDENT_COURSE_PK PRIMARY KEY (
STUDENT_ID, COURSE_ID );

ALTER TABLE AD_STUDENT_COURSE_DETAILS ADD CONSTRAINT AD_STUDENT_COURSE_FK1 FOREIGN KEY (
STUDENT_ID ) REFERENCES AD_STUDENTS ( STUDENT_ID ) ;

ALTER TABLE AD_STUDENT_COURSE_DETAILS ADD CONSTRAINT AD_STUDENT_COURSE_FK2 FOREIGN KEY (
COURSE_ID ) REFERENCES AD_COURSES ( COURSE_ID ) ;

ALTER TABLE AD_FACULTY_COURSE_DETAILS ADD CONSTRAINT AD_FACULTY_COURSE_PK PRIMARY KEY (
FACULTY_ID, COURSE_ID ) ;
```

```
ALTER TABLE AD_FACULTY_COURSE_DETAILS ADD CONSTRAINT AD_FACULTY_COURSE_FK1 FOREIGN KEY (
FACULTY_ID ) REFERENCES AD_FACULTY ( FACULTY_ID ) ;

ALTER TABLE AD_FACULTY_COURSE_DETAILS ADD CONSTRAINT AD_FACULTY_COURSE_FK2 FOREIGN KEY (
COURSE_ID ) REFERENCES AD_COURSES ( COURSE_ID ) ;

ALTER TABLE AD_FACULTY_LOGIN_DETAILS ADD CONSTRAINT AD_FACULTY_LOGIN_PK PRIMARY KEY (
FACULTY_ID, LOGIN_DATE_TIME );

ALTER TABLE AD_FACULTY_LOGIN_DETAILS ADD CONSTRAINT AD_FACULTY_LOGIN_FK FOREIGN KEY (
FACULTY_ID ) REFERENCES AD_FACULTY ( FACULTY_ID ) ;
```

2. Alter the table `AD_FACULTY_LOGIN_DETAILS` and specify a default value for the column `LOGIN_DATE_TIME` of `SYSDATE`.

Solution:

```
ALTER TABLE AD_FACULTY_LOGIN_DETAILS MODIFY (LOGIN_DATE_TIME TIMESTAMP DEFAULT SYSDATE);
```

3. Set the `AD_PARENT_INFORMATION` table to a read-only status.

Solution:

```
ALTER TABLE AD_PARENT_INFORMATION READ ONLY;
```

NOTE: You can execute the `INSERT / ALTER TABLE` statements in Oracle Application Express in one of the two ways:

Method 1:

a. Open *Oracle Application Express* and paste the commands into the SQL Commands screen one at a time and run.

Method 2:

a. Open *Oracle Application Express* and use the same script upload method as you did with the DDL commands above.

** Note : Completed script is available in Section 0 (Zip - Academic DB Script)

Exercise 3: Creating Composite Primary, Foreign and Unique Keys

Overview

In this practice, you will create:

- Composite Primary Key
- Composite Foreign Key
- Composite Unique Key

** **Note – these tables are not a part of the Academic Database**

Tasks

1. Create the `DEPT` table with the following structure:

Column	Data Type	Description
dept_id	number(8)	Department ID
dept_name	varchar2(30)	Department Name
loc_id	number(4)	Location ID

The primary key for this table needs to be defined as a composite comprising of the `dept_id` and `loc_id`.

Solution:

```
CREATE TABLE DEPT(
dept_id number(8),
dept_name varchar2(30),
loc_id number(4),
CONSTRAINT dept_loc_pk PRIMARY KEY(dept_id,loc_id));
```

2. Create the SUPPLIERS and PRODUCTS table with the following structure:

SUPPLIERS TABLE

Column	Data Type	Description
sup_id	number(15)	Supplier ID part of composite primary key
sup_name	varchar2(30)	Supplier Name part of composite primary key
contact_name	number(4)	Agent Contact Name

The primary key for this table needs to be defined as a composite comprising of the `sup_id` and `sup_name`.

PRODUCTS TABLE

Column	Data Type	Description
product_id	number(10)	Product ID is the primary key
sup_id	number(15)	Supplier ID that does not hold NULL value
sup_name	varchar2(30)	Supplier Name that does not hold NULL value

The primary key for this table is `product_id.` The foreign key for this table needs to be defined as a composite comprising of the `sup_id` and `sup_name`.

Solution:

```
CREATE TABLE SUPPLIERS
( sup_id NUMBER(15),
  sup_name VARCHAR2(45),
  contact_name VARCHAR2(45),
  CONSTRAINT sup_id_name_pk PRIMARY KEY (sup_id, sup_name)
);

CREATE TABLE PRODUCTS
( product_id numeric(10),
  sup_id NUMBER(15) NOT NULL,
   sup_name VARCHAR2(45) NOT NULL,
  CONSTRAINT product_pk PRIMARY KEY ( product_id ) ,
  CONSTRAINT sup_comp_fk  FOREIGN KEY (sup_id, sup_name) REFERENCES
  suppliers(sup_id, sup_name)
);
```

3. Create the DEPT_SAMPLE table with the following structure:

Column	Data Type	Description
dept_id	number(8)	Department ID
dept_name	varchar2(30)	Department Name
loc_id	number(4)	Location ID

The UNIQUE key for this table needs to be defined as a composite comprising of the dept_id and dept_name.

Solution:

```
CREATE TABLE DEPT_SAMPLE (
dept_id NUMBER(8),
dept_name VARCHAR2(30),
loc_id NUMBER(4),
CONSTRAINT dept_det_uk UNIQUE (dept_id, dept_name));
```

** Note : Completed script is available in Section 0 (Zip - Academic DB Script)