UNIVERSITY of the WESTERN CAPE

IFS244 Group Assignment

EMS
Where your Future is our Priority

Group name/number	Group 5
Date of submission	18 October 2023
Lecturer details	Dr. Agunbiade Olusanya Yinka

PLAGIARISM DECLARATION

1. We know that plagiarism is wrong. Plagiarism is the use of another's work without acknowledging the origin of the work, and pretending that it is one's own work.
2. We know that plagiarism is a punishable offence and that UWC has strict disciplinary measures against this.
3. We declare therefore, that each significant contribution to, and quotation in this document, that we have taken from the work(s) of other people have been correctly attributed and have been correctly cited (referenced) using the APA format.
4. We declare that this document is our own original work.
5. We have not allowed, and will not allow, anyone to copy our work with the intention of passing it off as his or her own work.

GROUP CONTRIBUTION MATRIX AND PLAGIARISM DECLARATION

#	Student number	Name	Surname	Signature	% Contribution
1.	4227160	Zuhar	Regal	Z.Regal	100
2.	4263636	Martha	Mhlanga	M.Gravel	100
3.	4238315	Mchele	Mkhombo	M.S.M	100
4.	4159413	Ghumairaa	Misbach	G.M	100
5.	4206385	Keenan Mornay	Engelbrecht		0

Introduction

Dynamic IT Group is a private company that has recently been awarded a project to develop a database for University of the Western Cape (UWC) student registration system. **Our task is to proceed with the development of this database project.**

Business rules:

- A student can make one and only one registration.
- One registration can be made to one and only one course
- Zero or many students can be registered to one course.
- A staff member can exist without being in an academic department.
- A registration can be made to one or more subjects.
- One or more courses can have one or more subjects.
- An academic staff member can exist without being assigned to a subject

---

Entity Description

These descriptions provide a clear understanding of each entity and its attributes within the database schema.

Registration (o_registration):

- This entity represents the records of students registering for courses within UWC. It tracks essential information about student registrations.
- Attributes:
    - `regid`: A unique numerical identifier for each registration.
    - `regdate`: The date when the registration was made.
    - `regstatus`: The status of the registration, indicating whether it's pending, approved, or another status.
    - `regfees`: The registration fees associated with the registration.
    - `studentid`: A reference to the student involved in the registration.
    - `courseid`: A reference the course being registered for.
    - `deptid`: A reference to the department to which the student is affiliated.

- `faculty id`: A reference to the faculty or staff member responsible for overseeing this registration.
- `payid`: A reference for the payment made for this registration.
- `appid`: A reference to an advising appointment associated with this registration.

Department (o_department):

- This entity represents academic departments within UWC
- Attributes:
  - `deptid`: A unique numerical identifier for each department.
  - `dname`: The name of the department.
  - `facultyid`: A reference to which faculty is affiliated with each department.

Faculty (o_faculty):

- This entity represents the academic faculty's within UWC.
- Attributes:
  - `facultyid`: A unique numerical identifier for each faculty.
  - `fname`: The name of the faculty.

Lecturer (o_lecturer):

- This entity represents individuals who teach subjects within the educational institution.
- Attributes:
  - `lecturerid`: A unique numerical identifier for each lecturer.`l_name`: The first name of the lecturer.
  - `l_lastname`: The last name of the lecturer.
  - `l_email`: The email address of the lecturer.
  - `subjectid`: A reference for the subject the lecturer teaches.
  - `deptid`: A reference to the department to which the lecturer is affiliated.

Course (o_course)

- This entity represents the courses offered at UWC that belong to various academic departments.
- Attributes:
    - `courseid`: A unique numerical identifier for each course.
    - `coursename`: The name of the course.
    - `deptid`: A reference to the academic department in which the course belongs.

Subject (o_subject)

- This entity represents the subjects offered at UWC within various courses.
- Attributes:
    - `subjectid`: A unique numerical identifier for each subject.
    - `subjectname`: The name of the subject.
    - `subjectcode`: The code of the subject, specific to UWC.

Course Subject (o_coursesubject)

- This entity represents the many-to-many relationship between students and courses.
- This allows multiple students to be registered for the same course.
- Attributes:
    - `coursesubjectid`: A unique numerical identifier for each course-subject relationship.
    - `courseid`: A reference to a specific course.
    - `subjectid`: A reference to a specific subject

Payment (o_payment)

- This entity represents the financial transactions within the university. Helping them track and manage financial matters effectively.
- Attributes :
    - `payid`: a numerical identifier of each payment
    - `payamount`:a numeric value that represents the amount of money paid for a specific transaction within the university (tution fees)
    - `paydate`: The date on which the transaction/payment was made

        - `paymethod`: The method or mode used to make the transaction/payment
        - `paybranch`: Represents the particular bank used to make the transaction/payment

Advisor (o_advisor)

- This entity represents a numerical identifier of an advisor
- Attributes
    - `avid`: Reference number assigned to each student advisor
    - `advname`: Represents the name of the student advisor
    - `advsurname`: The surname of the student advisor
    - `advemail`: Represents the email of the student advisor
    - `advphone`: Represents the phone number of the student advisor
    - `appid`: A unique reference number assigned to each advising appointment

Advising Appointment (advising_app)

- This entity represents information about scheduled advising appointments between students and their academic advisors
- Attributes
    - `appid`: A unique reference number assigned to each advising appointment
    - `studentid`: Represents the unique identifier for the student who scheduled the advising appointment
    - `appdate`: The date of the advising appointment
    - `apptime`: The time at which the advising appointment is scheduled
    - `applocation`: The physical location where the advising appointment will take place

Student(o_student)

- This entity represents the students who are registered to the academic institution. It stores essential student data.
- Attributes:

- `studentid`: A unique numerical identifier for each student
- `s_name`: The first name of the student
- `s_lastname`: The last name of the student
- `s_gender`: the gender/sex of the student
- `s_date_of_birth`: The date of birth of the student
- `s_address`: The student's residential address
- `s_city`: The city the student is from
- `s_emergency_contact`: The student's emergency contact
- `s_phone`: The student's personal phone number

Results(o_results)

- This entity represents the results that each student gets for their respective course.
- Attributes:
    - `pastresultid`: A unique numerical identifier for a student's results
    - `studentid`: A unique numerical identifier for each student
    - `coursid`: A unique numerical identifier for each course
    - `course_mark`: The final result that a student gets for a course

StudentSubject(o_studentsubject)

- This entity represents a bridge table between the student table (o_student) and the subject table (o_subject).
- Attributes:
    - `studentid`: A unique numerical identifier for each student
    - `subjectid`: A unique numerical identifier for each subject

StudentCourse (o_studentcourse)

- This entity represents a bridge table between the student table (o_student) and course table (o_course).
- Attributes:
    - `studentid`: A unique numerical identifier for each student
    - `courseid`: A unique numerical identifier for each course

Alignment with business rules

1. A student can make one and only one registration:

    - The `o_registration` table includes 'studentid' as a foreign key, ensuring that each registration is associated with one student. This aligns with the rule.

2. One registration can be made for one and only one course:

    - The `o_registration` table includes `courseid` as a foreign key, ensuring that each registration is associated with one course. This aligns with the rule.

3. Zero or many students can be registered for a course:

    - The ‘o_studentcourse’ table is responsible for managing the many-to-many relationship between students and courses.It becomes possible to register zero, one, or many students for a particular course, which aligns with the specified rule.

4. A staff member can exist without being in an academic department:

    - The `o_lecturer` table includes `deptid` as a foreign key but does not make it mandatory. Therefore, a staff member can exist without being linked to a department, aligning with the rule.

5. A registration can be made for one or more subjects:

    - The `o_registration` table includes `subjectid` as a foreign key, ensuring that each registration is associated with one or more subjects. This aligns with the rule.

6. One or more courses can have one or more subjects:

    - The `o_coursesubject` table allows for a many-to-many relationship between courses and subjects, supporting this rule.

7. An academic staff member can exist without being assigned to a subject:

- The `o_lecturer` table includes a `subjectid` foreign key but does not make it mandatory. This aligns with the rule, allowing staff members to exist without subjects.

Datatypes for entities

- Ensuring Data integrity

o_department	
deptid	NUMBER(6) NOT NULL, PK
dname	VARCHAR(50)
facultyid	NUMBER(6) NOT NULL, FK

o_registration	
regid	NUMBER(6) NOT NULL, PK
regdate	DATE
regstatus	VARCHAR2(20)
regfees	NUMBER(7,2)
studentid	NUMBER(6) NOT NULL, FK
courseid	NUMBER(6) NOT NULL, FK
deptid	NUMBER(6) NOT NULL, FK
facultyid	NUMBER(6) NOT NULL, FK

payid	NUMBER(6) NOT NULL, FK
appid	NUMBER(6) NOT NULL, FK

o_faculty	
facultytid	NUMBER(6) NOT NULL, PK
fname	VARCHAR(50)

o_lecturer	
lecturerid	NUMBER(6) NOT NULL, PK
l_name	VARCHAR2(20)
l_lastname	VARCHAR2(20)
l_email	VARCHAR2(50),
subjectid	NUMBER(6)NOT NULL,FK
deptid	NUMBER(6)NOT NULL,FK

o_student	
studentid	NUMBER(6) NOT NULL, PK

s_name	VARCHAR2(50)
s_lastname	VARCHAR2(50)
s_email	VARCHAR2(100)
s_gender	CHAR(1)
s_date_of_birth	DATE
s_address	VARCHAR2(200)
s_city	VARCHAR2(50)
s_emergency_contact	VARCHAR2(12)
s_phone	VARCHAR2(12)

o_results	
pastresultid	NUMBER(6) NOT NULL, PK
studentid	NUMBER(6) NOT NULL, FK
courseid	NUMBER(6) NOT NULL, FK
course_mark	VARCHAR(4)

o_studentsubject	
studentid	NUMBER(6), FK
subjectid	NUMBER(6), FK

o_studentcourse	
studentid	NUMBER(6), FK
courseid	NUMBER(6), FK

o_course	
courseid	NUMBER(6), PK
coursename	VARCHAR(20)
deptid	NUMBER(6), FK

o_subject	
subjectid	NUMBER(6) NOT NULL, PK
subjectname	VARCHAR(40)
subjectcode	VARCHAR(6)

o_coursesubject	
coursesubjectid	NUMBER(6) NOT NULL, PK
courseid	NUMBER(6) NOT NULL, FK
subjectid	NUMBER(6) NOT NULL, FK

o_payment	
payid	NUMBER(6) NOT NULL, PK
payamount	NUMBER
paydate	DATE
paymethod	VARCHAR2(20)
paybranch	VARCHAR2(25)

o_advisor	
advtid	NUMBER(6) NOT NULL, PK
advname	VARCHAR2(20)
advsurname	VARCHAR2(20)
advemail	VARCHAR2(25)
advphone	VARCHAR2(25)
appid	NUMBER(6)NOT NULL, FK

o_advisingapp

appid	NUMBER(6), PK
studentid	VARCHAR(20), FK
appdate	DATE
apptime	TIME
applocation	VARCHAR2(25)

ERD for Registration Database

1. One-to-One Relationship: Student to Registration:

- Each student can have one and only one registration. This relationship is represented by the StudentID foreign key in the o_registration table.

2. One-to-One Relationship: Course to Registration:

- One registration can be made for one and only one course. This relationship is represented by the CourseID foreign key in the o_registration table.

3. Many-to-Many Relationship: Student to Course:

- Zero or many students can be registered for a course. This relationship is managed through the o_studentcourse table, which links students (StudentID) to courses (CourseID).

4. Optional Relationship: Lecturer to Department:

- A staff member can exist without being in an academic department. This is achieved by including DeptID as a foreign key in the o_lecturer table but not making it mandatory. Therefore, staff members can exist without being assigned department to a department.

5. One-to-Many Relationship: Registration to Subject:

- A registration can be made for one or more subjects. This is represented by the SubjectID foreign key in the o_registration table.

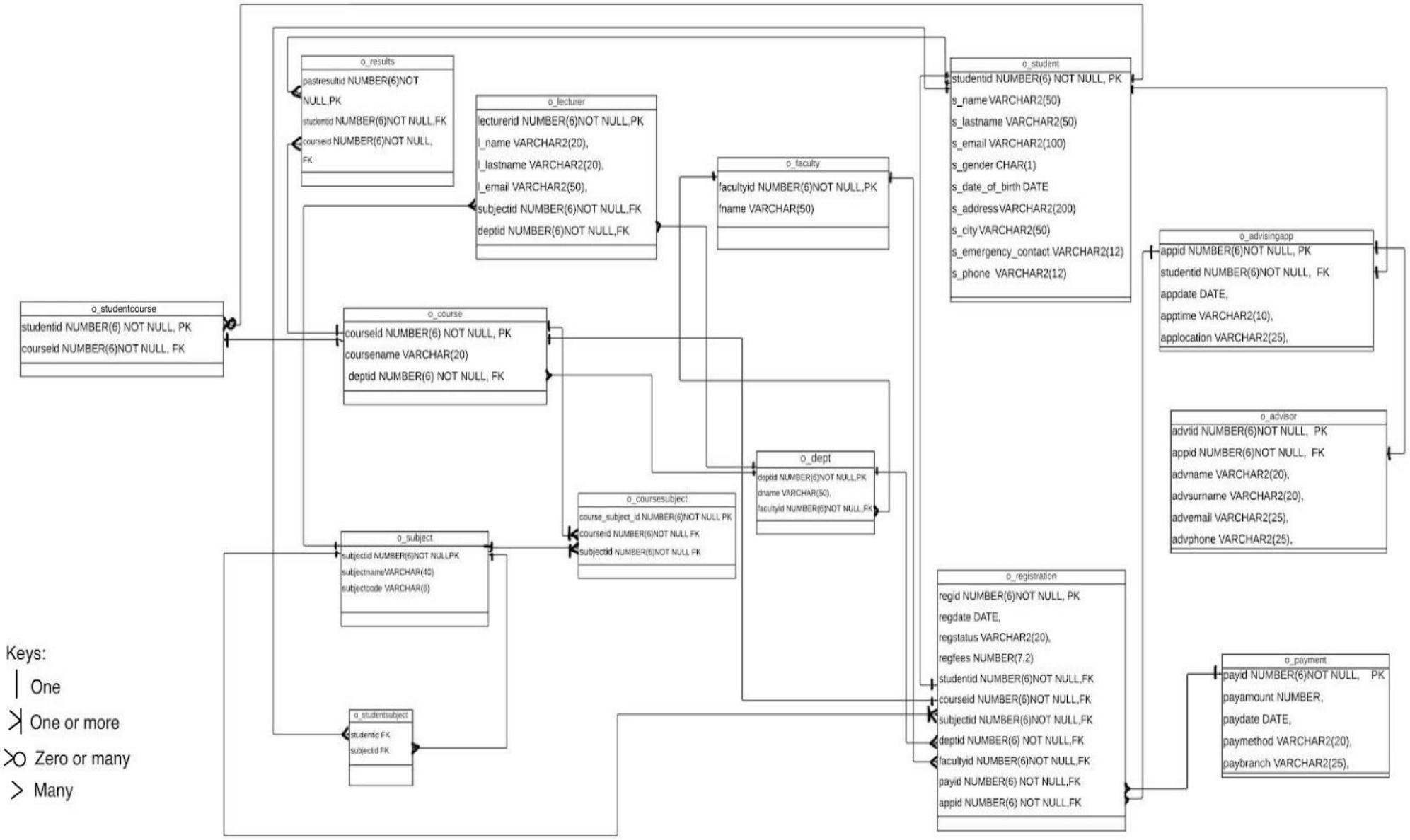
6. Many-to-Many Relationship: Course to Subject:

- One or more courses can have one or more subjects. This relationship is managed through the o_coursesubject table, which links courses (CourseID) to subjects (SubjectID).

7. Optional Relationship: Lecturer to Subject:

- An academic staff member can exist without being assigned to a subject. This is accomplished by including SubjectID as a foreign key in the o_lecturer table but not making it mandatory. Thus, academic staff members can exist without subject being assigned to a subject.

The ERD visually represents how these tables are related and aligned with the business rules. It illustrates the entities, attributes, and the nature of the relationships between them, providing a clear view of how data is structured in the database while ensuring compliance with the rules.


o_results
pastresultid NUMBER(6)NOT NULL,PK
studentid NUMBER(6)NOT NULL,FK
courseid NUMBER(6)NOT NULL, FK
o_lecturer
lecturerid NUMBER(6)NOT NULL,PK
l_name VARCHAR2(20),
l_lastname VARCHAR2(20),
l_email VARCHAR2(50),
subjectid NUMBER(6)NOT NULL,FK
deptid NUMBER(6)NOT NULL,FK
o_faculty
facultyid NUMBER(6)NOT NULL,PK
fname VARCHAR(50)
o_student
studentid NUMBER(6) NOT NULL, PK
s_name VARCHAR2(50)
s_lastname VARCHAR2(50)
s_email VARCHAR2(100)
s_gender CHAR(1)
s_date_of_birth DATE
s_address VARCHAR2(200)
s_city VARCHAR2(50)
s_emergency_contact VARCHAR2(12)
s_phone VARCHAR2(12)
o_advisingapp
appid NUMBER(6)NOT NULL, PK
studentid NUMBER(6)NOT NULL, FK
appdate DATE,
apptime VARCHAR2(10),
applocation VARCHAR2(25),
o_studentcourse
studentid NUMBER(6) NOT NULL, PK
courseid NUMBER(6)NOT NULL, FK
o_course
courseid NUMBER(6) NOT NULL, PK
coursename VARCHAR(20)
deptid NUMBER(6) NOT NULL, FK
o_dept
deptid NUMBER(6)NOT NULL,PK
dname VARCHAR(50),
facultyid NUMBER(6)NOT NULL,FK
o_advisor
advtid NUMBER(6)NOT NULL, PK
appid NUMBER(6)NOT NULL, FK
advname VARCHAR2(20),
advsurname VARCHAR2(20),
advemail VARCHAR2(25),
advphone VARCHAR2(25),
o_coursesubject
course_subject_id NUMBER(6)NOT NULL PK
courseid NUMBER(6)NOT NULL FK
subjectid NUMBER(6)NOT NULL FK
o_subject
subjectid NUMBER(6)NOT NULL,PK
subjectnameVARCHAR(40)
subjectcode VARCHAR(6)
o_registration
regid NUMBER(6)NOT NULL, PK
regdate DATE,
regstatus VARCHAR2(20),
regfees NUMBER(7,2)
studentid NUMBER(6)NOT NULL,FK
courseid NUMBER(6)NOT NULL,FK
subjectid NUMBER(6)NOT NULL,FK
deptid NUMBER(6) NOT NULL,FK
facultyid NUMBER(6)NOT NULL,FK
payid NUMBER(6) NOT NULL,FK
appid NUMBER(6) NOT NULL,FK
o_payment
payid NUMBER(6)NOT NULL, PK
payamount NUMBER,
paydate DATE,
paymethod VARCHAR2(20),
paybranch VARCHAR2(25),
o_studentsubject
studentid FK
subjectid FK
Keys:
One
One or more
Zero or many
Many