From: "Firdavs Zaynidinov" <mirzo.zaynidinov@gmail.com>
To: "Firdavs Zaynidinov" <mirzo.zaynidinov@gmail.com>
Date: 11/4/2024 12:33:18 AM
Subject: Fwd: Базы данных. ДЗ-2. Моделирование

---------- Forwarded message ---------
От: Google Формы <forms-receipts-noreply@google.com>
Date: пн, 23 сент. 2024 г. в 03:12
Subject: Базы данных. ДЗ-2. Моделирование
To: <mirzo.zaynidinov@gmail.com>

Благодарим за заполнение формы Базы данных. ДЗ-2. Моделирование!

Полученные ответы

Изменить ответ

Базы данных. ДЗ-2. Моделирование

Моделирование БД «Университет»

Электронная почта *

mirzo.zaynidinov@gmail.com

Студент *

Зайнидинов Мирзофирдавс Шавкатович

0. Диаграммы ERM и PDM *

Картинка размером не больше 1920x1080. В левой половине — ERM. В правой — PDM. Для удобства чтения таблицы должны находиться на тех же местах, что и сущности/связи/ассоциации, из которых они получены.

Шрифт размером не менее 10pt. Рекомендуется использовать прямые линии (а не дуги) и избегать пересечений.

Отправленные файлы

Screenshot 2024-09-23 at 03.05.59 - Firdavs Zaynidinov.jpg

1. Модель сущность-связь

1.0. Нотация *

Укажите тип используемой нотации, в том числе (не)обязательность по умолчанию для атрибутов и связей.

Нотация Crows foot, атрибуты по умолчанию обязательны, связи необязательны

1.1. Сущности и атрибуты *

Опишите выделенные сущности (в том числе, слабые), их атрибуты.

Сущность Faculties, используется для хранения информации о факультетах, атрибуты:
facluty_id - идентификатор факультета, первичный ключ.
faculty_name - наименования факультета.

Сущность Groups, используется для хранения информации о группах, атрибуты:
group_id - идентификатор группы, первичный ключ.
group_name - наименования группы

Сущность Students, используется для хранения информации о студентах, атрибуты:
student_id - идентификатор студента, первичный ключ.
student_fullname - ФИО студента.
group_id - идентификатор группы в котором учиться студент.
email_address - электронная почта студента.

Сущность Professors, используется для хранения информации о преподавателях, атрибуты:
professor_id - идентификатор преподавателя, первичный ключ.
professor_fullname - ФИО преподавателя.
email_address - электронная почта преподавателя

Сущность Subjects, используется для хранения информации о предметах, атрибуты:
subject_id - идентификатор предмета.

subject_name - название предмета.

1.2. Связи и ассоциации *

Опишите выделенные связи и ассоциации.

Связь между Faculties и Groups: один ко многим, причем в факультете может не быть групп, но группа обязательно должен быть в факультете.

Связь между Groups и Students: один ко многим, причем в группе может не быть студентов, но студент обязательно должен быть в группе.

Связь между группой, преподавателей и предметов через ассоциацию LinkProfessorWithGroupAndSubject нужна чтобы узнать какой преподаватель ведет конкретный предмет в конкретной группе. Каждый преподаватель может преподавать в нескольких группах по несколько предметов и у каждого предмета есть преподаватель.

Связь между Subjects и Students через ассоциацию Marks, у студентов может не быть или быть оценка по любому предмету.

2. Физическая модель

2.0. Нотация *

Укажите тип используемой нотации, в том числе (не)обязательность по умолчанию для столбцов.

Нотация графическая, столбцы по умолчанию обязательны,

2.1. Домены *

Опишите преобразование доменов, по одному на строке в формате

«домен → тип» .

В этом и последующих домашних заданиях:
- кавычки не являются частью синтаксиса
- можно вместо
«→»
использовать

«->» .

Например,
hello_world_message -> char(12)
hello_world_index -> numeric(3)

id -> int
name -> varchar(100)
value -> varchar(5)
email -> varchar(50)
s_name -> varchar(50)
group_name -> varchar(10)

2.2. Преобразование *

Опишите особенности преобразования (в частности, для слабых сущностей и ассоциаций).

По сущностям:
Сущности преобразовались а таблицы атрибуты преобразовались в столбцы, домены превратились в типы, ключи остались ключами.

По ассоциациям:
Ассоциация Marks преобразилась в таблицу Marks у которого состовной ключ (student_id, group_id, subject_id) из которого мы для заданного студента из какой-то группы по заданному предмету можем узнать его оценку. Оценку можно поставить тогда и только тогда если у студента из какой-то группы есть этот предмет.

Ассоциация LinkProfessorWithSubjectAndGroup преобразовалась в таблицу LinkProfessorWithSubjectAndGroup, у которого составной ключ (professor_id, group_id, subject_id), с помощью которого мы можем найти преподавателя предмета какой-то группы.

3. SQL

Помните, что SQL – это язык программирования, и не забывайте о форматировании и отступах.

3.1. DDL *

Схема базы данных.

```
create table Faculties
(
    faculty_id        int         primary key,
    faculty_name varchar(100) not null
);

create table Groups
(
    group_id      int    primary key,
    group_name  varchar(10)  not null,
    faculty_id          int                 not null
);
```

```
alter table Groups
    add foreign key (faculty_id)
        references Faculties (faculty_id);

create table Subjects
(
    subject_id        int     primary key,
    subject_name varchar(50)  not null
);

create table Students
(
    student_id    int           primary key,
    student_fullname   varchar(100)  not null,
    group_id      int                        not null,
    email_address       varchar(100)         not null
);

alter table Students
    add foreign key (group_id)
        references Groups (group_id);

alter table Students
    add constraint student_second_key
        unique (student_id, group_id);

create table Professors
(
    professor_id      int     primary key unique,
    professor_fullname  varchar(100)  not null,
    email_address         varchar(100)  not null
);

create table LinkProfessorWithSubjectAndGroup
(
    professor_id int not null,
    group_id        int not null,
    subject_id     int not null
);

alter table LinkProfessorWithSubjectAndGroup
    add foreign key (group_id)
        references Groups (group_id);

alter table LinkProfessorWithSubjectAndGroup
    add foreign key (subject_id)
        references Subjects (subject_id);
```

```
alter table LinkProfessorWithSubjectAndGroup
    add foreign key (professor_id)
        references Professors (professor_id);

alter table LinkProfessorWithSubjectAndGroup
    add constraint link_professor_with_subject_and_group_key
        unique (professor_id, group_id, subject_id);

create table Marks
(
    student_id   int          not null,
    subject_id   int          not null,
    group_id           int          not null,
    mark               varchar(5)  not null
);

alter table Marks
    add foreign key (student_id, group_id)
        references Students (student_id, group_id);

alter table Marks
    add foreign key (subject_id)
        references Subjects (subject_id);

alter table Marks
    add constraint marks_key
        unique (student_id, subject_id, group_id);
```

3.2. DML *

Пример тестовых данных. Достаточно 2-3 записей на таблицу, если они в полной мере демонстрируют особенности БД.

Для удобства чтения данные должны быть записаны в том же порядке, что и в схеме.

```
INSERT INTO FACULTIES
 (faculty_id, faculty_name,)
VALUES
    (1, 'ФИТиП'),
    (2, 'ФПИиКТ'),
    (3, 'ФИКТ'),
    (4, 'ФКТиУ'),
    (5, 'ФФиОИ');

INSERT INTO Groups
    (group_id, group_name, faculty_id)
VALUES
    (1, 'M3436', 1),
```

```
    (2, 'M3437', 1),
    (3, 'M3438', 1),
    (4, 'A4132', 4),
    (5, 'C3435', 5);

INSERT INTO Subjects
 (subject_id, subject_name)
VALUES
    (1, 'Математическая логика'),
    (2, 'Математический анализ семестр 1'),
    (3, 'Математический анализ семестр 2'),
    (4, 'Компьютерные сети'),
    (5, 'Машинное обучение');

INSERT INTO Students
    (student_id, student_fullname, group_id, email_address)
VALUES
    (1, 'Студент1', 2, 'student1@gmail.com'),
    (2, 'Студент2', 1, 'student2@gmail.com'),
    (3, 'Студент3', 1, 'student@three.ru'),
    (4, 'Студент4', 3, 'student4@mail.ru'),
    (5, 'Студент5', 4, 'student5@gmail.com');

INSERT INTO Professors
    (professor_id, professor_fullname, email_address)
VALUES
    (1, 'Proffesor1', 'pr0fessor1@gmail.com'),
    (2, 'Professor2', 'pr0fessor2@mail.ru'),
    (3, 'Professor3', 'pr0fessor@three.ru');

INSERT INTO LinkProfessorWithSubjectAndGroup
    (professor_id, group_id, subject_id)
VALUES
    (1, 4, 2),
    (2, 5, 3),
    (3, 1, 4);

INSERT INTO Marks
    (student_id, subject_id, group_id, mark)
VALUES
    (1, 5, 2, 'A'),
    (2, 4, 1, 'B'),
    (3, 3, 1, 'C'),
    (4, 2, 3, 'D'),
    (5, 1, 4, 'E');
```