1. Для заданной предметной области спроектировать базу данных с использованием правил **нормализации.** Представить схему данных (ER-диаграмму).

2. Написать следующие запросы

* Получить список всех общеобразовательных дисциплин в учебном плане заданной специальности.
* Получить список специальностей, для которых преподается заданный предмет.
* Получить список преподавателей и недельной зарплаты каждого.
* Получить преподавателя с максимальной недельной зарплатой.
* Поучить учебный план заданной специальности: дисциплина, вид дисциплины, часов в неделю.

3. Первое правило нормализации. Определение. Пример.

Вид дисциплины зависит от названия дисциплины. Один и тот же преподаватель может вести разные предметы и преподавать на различных специальностях, но его должность при этом не меняется, то есть является его coбственым атрибутом. Почасовая оплата определяется только должностью.

Базу данных спроектировать и заполнить в СУБД MS Access. После заполнения каждая таблица должна содержать от-5 до-10 строчек. В отчет небходимо включить как тексты запросов, так и результаты их выполнения.

1. По заданию, в среде MS Access спроектирована реляционная база данных, ER-диаграмма которой приведена на рис. 1.

Рис. 1. ER-диаграмма базы данных

В таблице DISCIPL\_TYPE указывается вид дисциплины. Таблица DISCIPL содержит наименование дисциплины, и, кроме того, связана с таблицей DISCIPL\_TYPE. Таким образом, с каждой дисциплиной ассоциируется ее вид. Вид дисциплины был выделен в отдельную таблицу для того, чтобы при заполнении таблицы DISCIPL не было необходимости писать вручную вид дисциплины, ведь при этом легко ошибиться (много букв), а такая ошибка приведет к невозможности корректного выполнения некоторых запросов, например запроса №1 из задания. Таблица PREPOD служит для связи таблицы DOLJNOST, хранящей информацию о названии должности и соответствующей зарплате, с конкретным преподавателем. Таблица PLAN служит для связи между собой таблиц SPEC (хранит названия специальностей), DISCIPL, PREPOD. Кроме того, в таблице PLAN хранится число часов, которые отведены в неделю для занятий данной специальности по данной дисциплине с привлечением данного преподавателя. Такая структура учебного плана объясняется тем, что любой преподаватель может преподавать разные предметы на разных специальностях, либо одни и те же предметы на разных специальностях, либо разные предметы на одной специальности, и так далее в любой комбинации (что и наблюдается на практике). При этом их часовая ставка, согласно заданию, зависит от должности, но не от специальности или предмета. Количество часов, выданных данным преподавателем студентам данной специальности по данному предмету зависит только от учебного плана.

Таким образом, имеем нормализованную реляционную базу данных, в которой очень легко производить изменения. Так, например, если некий преподаватель уволился и вместо него пришел другой преподаватель и взял на себя все те же самые его часы, то достаточно (если нет необходимости в сохранении записей уволенного преподавателя) изменить имя ушедшего преподавателя на имя нового. Изменение названия специальности (например, если «Госуправление» переименуют в «Муниципальное управление») не отразится на связях в таблице, изменится лишь название специальности в таблице SPEC. Изменение наименования дисциплины, либо изменение ее вида опять же не приводит к существенным изменениям. Централизованное повышение почасовой ставки сразу всем сотрудникам, занимающим одну должность, производится через таблицу DOLJNOST. Используя таблицу PLAN мы можем очень быстро (и не затрагивая содержимого остальных таблиц) изменить учебный план, добавить новые часы, удалить исключенные из плана занятия, сменить преподавателя, изменить число часов.

Ниже представлено содержимое таблиц:

DISCIPL\_TYPE

DISCIPL

DOLJNOST

PREPOD

SPEC

PLAN

2. Согласно заданию были разработаны следующие запросы на языке SQL:

А) Получить список всех общеобразовательных дисциплин в учебном плане заданной специальности.

SELECT [discipl].[nazvanie]

FROM (discipl INNER JOIN discipl\_type ON [discipl].[id\_discipl\_type]=[discipl\_type].[Код]) INNER JOIN (spec INNER JOIN plan ON [spec].[Код]=[plan].[id\_spec]) ON [discipl].[Код]=[plan].[id\_discipl]

WHERE [spec].[nazvanie]=[?] And [discipl\_type].[nazvanie]='Общеобразовательные';

При исполнении данного запроса в среде MS Access будет запрошено название специальности (параметр ? ), в результате будет получен список общеобразовательных предметов в учебном плане данной специальности. Результат запроса для специальности «Автоматизация техпроцессов и производств» представлен ниже:

Б) Получить список специальностей, для которых преподается заданный предмет.

SELECT spec.nazvanie

FROM discipl INNER JOIN (spec INNER JOIN plan ON spec.Код = plan.id\_spec) ON discipl.Код = plan.id\_discipl

WHERE (discipl.nazvanie)=[?];

При исполнении данного запроса в среде MS Access будет запрошено название предмета (параметр ? ), в результате будет получен список специальностей, для которых преподается заданный предмет. Результат запроса для предмета «Философия» представлен ниже:

В) Получить список преподавателей и недельной зарплаты каждого.

SELECT prepod.name, sum(plan.number\_of\_hours\_at\_week\*doljnost.oplata\_za\_chas) AS oplata\_za\_week

FROM doljnost INNER JOIN (prepod INNER JOIN plan ON prepod.Код = plan.id\_prepod) ON doljnost.Код = prepod.id\_doljnost

GROUP BY prepod.name;

Результат запроса представлен ниже:

Г) Получить преподавателя с максимальной недельной зарплатой.

SELECT TOP 1 [prepod].[name], sum([plan].[number\_of\_hours\_at\_week]\*[doljnost].[oplata\_za\_chas]) AS oplata\_za\_week

FROM doljnost INNER JOIN (prepod INNER JOIN plan ON [prepod].[Код]=[plan].[id\_prepod]) ON [doljnost].[Код]=[prepod].[id\_doljnost]

GROUP BY [prepod].[name]

ORDER BY sum([plan].[number\_of\_hours\_at\_week]\*[doljnost].[oplata\_za\_chas]) DESC;

Результат запроса представлен ниже:

Д) Получить учебный план заданной специальности: дисциплина, вид дисциплины, часов в неделю.

SELECT [discipl].[nazvanie], [discipl\_type].[nazvanie], [plan].[number\_of\_hours\_at\_week]

FROM spec INNER JOIN ((discipl\_type INNER JOIN discipl ON [discipl\_type].[Код]=[discipl].[id\_discipl\_type]) INNER JOIN plan ON [discipl].[Код]=[plan].[id\_discipl]) ON [spec].[Код]=[plan].[id\_spec]

WHERE [spec].[nazvanie]=[?];

При исполнении данного запроса в среде MS Access будет запрошено название специальности (параметр ?). Результат запроса для специальности «Автоматизация техпроцессов и производств» представлен ниже:

3. Первое правило нормализации.

Таблица находится в первой нормальной форме, если каждый её атрибут атомарен. Под выражением «атрибут атомарен» понимается, что атрибут может содержать только одно значение. Таким образом, не соответствуют 1НФ таблицы, в полях которых могут храниться списки значений. Для приведения таблицы к 1НФ обычно требуется разбить таблицу на несколько отдельных таблиц.

Замечание: в реляционной модели отношение всегда находится в 1 (или более высокой) нормальной форме в том смысле, что иные отношения не рассматриваются в реляционной модели. То есть само определение понятия отношение заведомо подразумевает наличие 1НФ.

Вопрос об атомарности атрибутов решается на основе семантики данных, то есть их смыслового значения. Атрибут атомарен, если его значение теряет смысл при любом разбиении на части или переупорядочивании. И наоборот, если какой-либо способ разбиения на части не лишает атрибут смысла, то атрибут неатомарен.

**Пример приведения таблицы к первой нормальной форме**

Исходная, ненормализованная, таблица:

|  |  |
| --- | --- |
| **Сотрудник** | **Номер телефона** |
| Иванов И. И. | 31-56-8239-57-34 |
| Петров П. П. | 22-62-34 |

Таблица, приведённая к 1НФ:

|  |  |
| --- | --- |
| **Сотрудник** | **Номер телефона** |
| Иванов И. И. | 31-56-82 |
| Иванов И. И. | 39-57-34 |
| Петров П. П. | 22-62-34 |