Шубин Дмитрий

Проф. задания для кандидатов на прикладную разработку (C#)

Разработка на SQL.

## Вариант 1. Проектирование баз данных.

Разработать структуру базы данных (перечень таблиц с описанием полей, ключей, связей между таблицами) для учета проводимых курсов обучения.

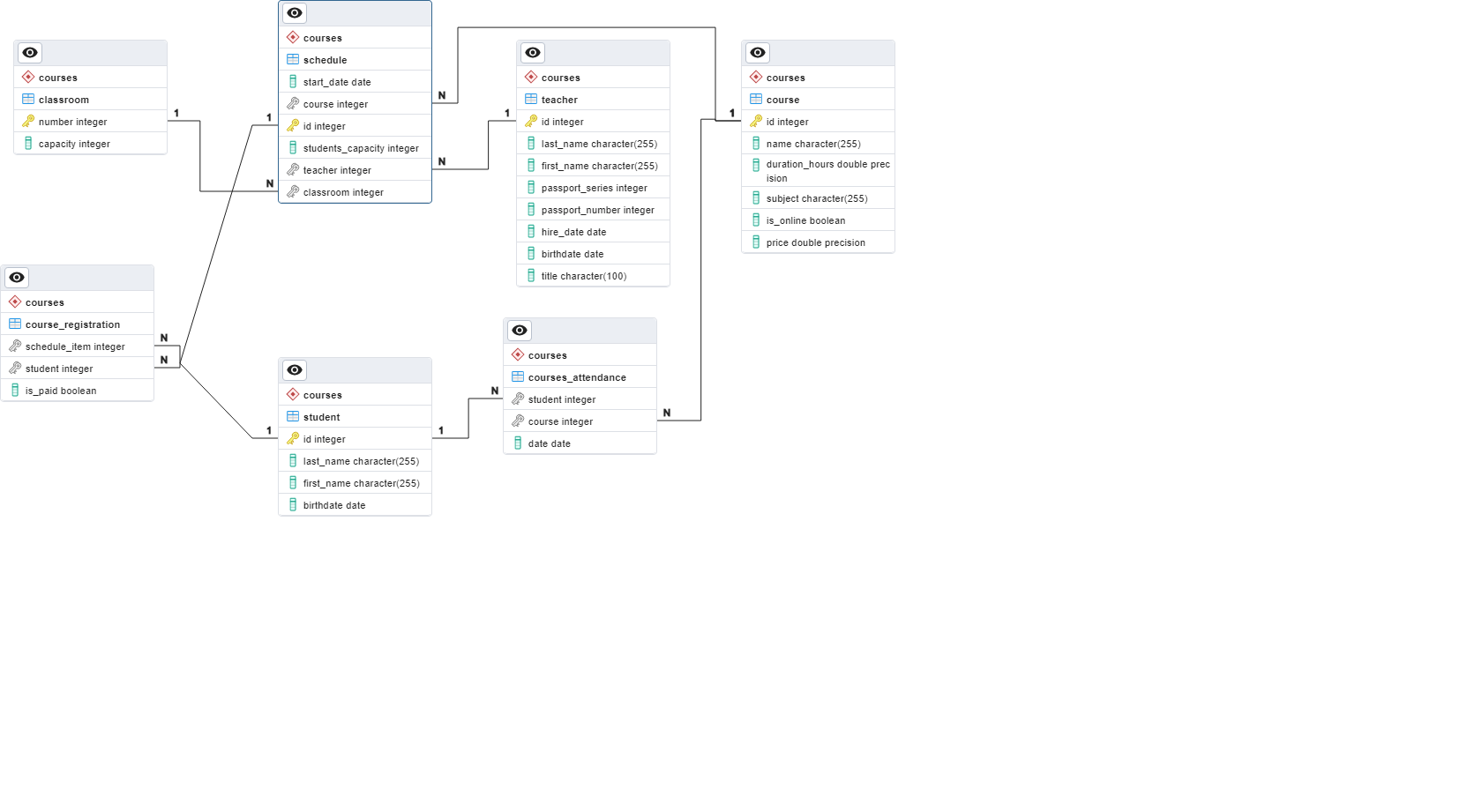


Рисунок 1. Схема БД курсов обучения

Таблица *student* (обучаемые)

|  |  |  |
| --- | --- | --- |
| **Колонка** | **Тип данных** | **Комментарий** |
| id | Целое число | Первичный ключ. Уникальный идентификатор |
| last\_name | Строка | Фамилия |
| first\_name | Строка | Имя |
| birthdate | Дата | Дата рождения |
| *При необходимости добавляются Отчество, Пол, Паспортные данные, Адрес, Телефон и т.д.* | | |

Таблица *course* (курсы)

|  |  |  |
| --- | --- | --- |
| **Колонка** | **Тип данных** | **Комментарий** |
| id | Целое число | Первичный ключ. Уникальный идентификатор |
| name | Строка | Название курса |
| duration\_hours | Десятичное число | Продолжительность курса в часах |
| subject | Строка | Предмет курса (напр. «астрономия») |
| is\_online | Булевый/логический | Курс онлайн/офлайн |
| price | Десятичное число | Стоимость прохождения курса |

Таблица *teacher* (преподаватели)

|  |  |  |
| --- | --- | --- |
| **Колонка** | **Тип данных** | **Комментарий** |
| id | Целое число | Первичный ключ. Уникальный идентификатор |
| last\_name | Строка | Фамилия |
| first\_name | Строка | Имя |
| passport\_series | Целое число | Серия паспорта. Число от 1000 до 9999 включительно. Проверяется при заполнении |
| passport\_number | Целое число | Номер паспорта. Число от 100000 до 999999 включительно. Проверяется при заполнении.  Комбинация серии и номера паспорта должны быть уникальны. Обеспечивается индексом. |
| hire\_date | Дата | Дата принятия на работу. |
| Birthdate | Дата | Дата рождения |
| Title | Строка | Наименование должности |
| *При необходимости добавляются Зарплата, Отчество, Пол и другие паспортные данные.* | | |

Таблица *classroom* (аудитории)

|  |  |  |
| --- | --- | --- |
| **Колонка** | **Тип данных** | **Комментарий** |
| number | Целое число | Первичный ключ. Номер аудитории |
| capacity | Целое число | Вместимость |
| *При необходимости добавляется информация о техническом оснащении аудиторий.* | | |

Таблица *schedule* (расписание)

|  |  |  |
| --- | --- | --- |
| **Колонка** | **Тип данных** | **Комментарий** |
| id | Целое число | Первичный ключ. Уникальный идентификатор |
| start\_date | Дата | Дата и время начала курса |
| course | Целое число | Внешний ключ. Значение id из таблицы courses |
| classroom | Целое число | Внешний ключ. Значение number из таблицы classroom.  Для онлайн курсов не заполняется (NULL) |
| student\_capacity | Целое число | Количество мест на курсе. Заполняется автоматически при заполнении поля classroom значением capacity из таблицы classroom |
| teacher | Целое число | Внешний ключ. Значение id из таблицы teacher |

Таблица *course\_registration* (регистрации на курсы)

|  |  |  |
| --- | --- | --- |
| **Колонка** | **Тип данных** | **Комментарий** |
| schedule\_item | Целое число | Внешний ключ. Значение id из таблицы schedule |
| student | Целое число | Внешний ключ. Значение id из таблицы student |
| is\_paid | Булевый/логический | Курс оплачен/не оплачен |

Таблица *course\_attendance* (посещение курсов)

*т.к. регистрация ещё не означает посещение*

|  |  |  |
| --- | --- | --- |
| **Колонка** | **Тип данных** | **Комментарий** |
| course | Целое число | Внешний ключ. Значение id из таблицы course |
| student | Целое число | Внешний ключ. Значение id из таблицы student  Индекс обеспечивает уникальность пары student + course. Не позволяет посетить одному студенту один курс дважды. |
| date | Дата | Дата прохождения курса |
| *При необходимости добавляется информация о результате прохождения курса* | | |

## Вариант 2. Написание запроса к базе данных

1. Написать SQL-запрос, который возвращает объем продаж в количественном выражении в разрезе сотрудников за период с 01.10.2013 по 07.10.2013:

* Фамилия и имя сотрудника;
* Объем продаж сотрудника.

Данные отсортировать по фамилии и имени сотрудника.

**SELECT** **CONCAT(**"Sellers"**.**"Surname"**,** "Sellers"**.**"Name"**)** **AS** "Сотрудник"**,**

**SUM(**"Sales"**.**"Quantity"**)** **AS** "Объём продаж"

**FROM** sales**.**"Sales"**,** sales**.**"Sellers"

**WHERE** "Sales"**.**"IDSel" **=** "Sellers"**.**"ID"

**AND** "Sales"**.**"Date" **>=** '01.10.2013' **AND** "Sales"**.**"Date" **<=** '07.10.2013'

**GROUP** **BY** "Сотрудник"

**ORDER** **BY** "Сотрудник"**;**

1. На основании созданного в первом задании запроса написать SQL-запрос, который возвращает процент объема продаж в разрезе сотрудников и продукции за период с 01.10.2013 по 07.10.2013:

* Наименование продукции;
* Фамилия и имя сотрудника;
* Процент продаж сотрудником данного вида продукции (продажи сотрудника данной продукции/общее число продаж данной продукции).

В выборку должна попадать продукция, поступившая за период с 07.09.2013 по 07.10.2013.

Данные отсортировать по наименованию продукции, фамилии и имени сотрудника.

**SELECT** "Products"**.**"Name" **AS** "Наименование продукции"**,**

**CONCAT(**"Sellers"**.**"Surname"**,** "Sellers"**.**"Name"**)** **AS** "Сотрудник"**,**

**SUM(**"Sales"**.**"Quantity"**)** **\*** 100.0 **/**

**SUM(SUM(**"Sales"**.**"Quantity"**))**

**OVER** **(PARTITION** **BY** "Sales"**.**"IDProd"**)**

**AS** "Процент продаж продукции"

**FROM** sales**.**"Sales"**,** sales**.**"Products"**,** sales**.**"Sellers"

**WHERE** "Sales"**.**"IDProd" **=** "Products"**.**"ID"

**AND** "Sales"**.**"IDSel" **=** "Sellers"**.**"ID"

**AND** "Sales"**.**"Date" **>=** '01.10.2013'

**AND** "Sales"**.**"Date" **<=** '07.10.2013'

**AND** "Sales"**.**"IDProd" **=** **ANY** **(SELECT** "IDProd"

**FROM** sales**.**"Arrivals"

**WHERE** "Arrivals"**.**"Date" **>=** '07.09.2013'

**AND** "Arrivals"**.**"Date" **<=** '07.10.2013'**)**

**GROUP** **BY** "Sales"**.**"IDProd"**,** "Наименование продукции"**,** "Сотрудник"

**ORDER** **BY** "Наименование продукции"**,** "Сотрудник"**;**