



Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Робототехника и комплексная автоматизация»

КАФЕДРА «Системы автоматизированного проектирования»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОЙ РАБОТЕ по дисциплине
БАЗЫ ДАННЫХ

НА ТЕМУ:

Разработка информационной системы
«Рекрутинг (набор новых сотрудников).»

Студент : РК6-41
(Группа)

(Подпись, дата)

О.И.Синякова
(И.О.Фамилия)

Руководитель курсового проекта

(Подпись, дата)

Н.В.Пивоварова
(И.О.Фамилия)

2018 г.

Оглавление

1. Оглавление	2
2. Аннотация	3
3. Описание предметной области	4
4. Техническое задание	5
5. Разработка инфологической модели предметной области	6
6. Построение логической модели базы данных	9
7. Разработка оперативных запросов	10
8. Разработка стандартного статистического отчета	12
9. Разработка плана хранимой процедуры	13
10. Текст хранимой процедуры	14
11. Разработка приложения конечного пользователя	15
12. Заключение	17

Аннотация

Курсовая работа посвящена формированию базы данных для фирмы. В ходе выполнения работы по полученному у преподавателя заданию была разработана инфологическая модель, построена логическая модель, разработаны оперативные запросы и процедура для формирования ведомостей, а также был разработан и реализован пользовательский интерфейс.

Описание предметной области

Некоторая фирма имеет штатное расписание, в котором перечислены все имеющиеся должности (название должности), «вилка» оклада, т.е. минимально и максимально возможный оклады, код подразделения, к которому относится эта должность.

Каждый сотрудник фирмы занимает одну из должностей штатного расписания. О сотрудниках известны фамилия, дата рождения, адрес, образование, дата зачисления на должность и оклад, реально получаемый в пределах соответствующей «вилки» штатного расписания, а также дата увольнения, которая для работающих сотрудников равна NULL.

Периодически должности освобождаются по различным причинам (увольнение или перевод сотрудника, открытие новых должностей)

Во всех этих случаях объявляется новая вакансия на свободную должность. О вакансии известно, на какую должность она объявлена, дата открытия вакансии. Кроме того, предусмотрена дата закрытия вакансии, которая соответствует дате принятия на работу нового сотрудника на соответствующую должность.

Набор сотрудников на имеющиеся вакансии проходит на конкурсной основе. Кандидат обязан пройти собеседование.

Собеседования с кандидатами проводят сотрудники рекрутинга (отдел фирмы).

Рекрутеры являются сотрудниками фирмы. По каждой открытой вакансии рекрутеры фирмы ведут собеседования с кандидатами.

Для каждого кандидата фиксируется его фамилия, место жительства, возраст, пол. Один и тот же кандидат может проходить несколько собеседований на различные вакансии.

По каждому собеседованию фиксируется дата проведения, код рекрутера (равен его коду, как сотрудника фирмы) и полученная кандидатом оценка по 10-бальной шкале и реальный оклад, который обсуждался с кандидатом. Предусмотрена также отметка о результате. Если в итоге кандидат принят на вакансию, то для соответствующего собеседования проставляется признак «Yes», в противном случае - «No».

Техническое задание

Этап проектирования.

1. Разработать инфологическую модель предметной области. Письменно обосновать выделение сущностей и связей между ними, а также назначение атрибутов сущностей.
2. Разработать реляционную логическую модель будущей базы данных.
3. Разработать форму регулярного отчета, который позволит оценить эффективность соответствующего бизнеса.
4. Разработать план хранимой процедуры для реализации регулярного отчета.
5. Разработать интерфейс конечного пользователя для доступа к оперативным запросам и регулярному отчету.

Этап реализации.

1. Создать базу данных на основе разработанной логической модели.
2. Реализовать оперативные запросы, создав для отладки каждого из них тестовый набор данных в соответствующих таблицах.
3. Написать и отладить хранимую процедуру для реализации регулярного отчета.
4. Написать и отладить интерфейс конечного пользователя в форме меню, позволяющего выполнить любой запрос и получить регулярный отчет за любой период времени.

Разработка инфологической модели предметной области

Для создания инфологической модели предметной области используем метод выделения существительных из описания предметной области. Выделим в описании предметной области существительные жирным шрифтом.

Некоторая фирма имеет **штатное расписание**, в котором перечислены все имеющиеся **должности** (название должности), «вилка» оклада, т.е. **минимально и максимально** возможный **оклады**, **код подразделения**, к которому относится эта должность.

Каждый **сотрудник** фирмы занимает одну из должностей штатного расписания. О сотрудниках известны **фамилия**, **дата рождения**, **адрес**, **образование**, **дата зачисления** на должность и **оклад**, реально получаемый в пределах соответствующей «вилки» штатного расписания, а также **дата увольнения**, которая для работающих сотрудников равна NULL.

Периодически должности освобождаются по различным причинам (увольнение или перевод сотрудника, открытие новых должностей)

Во всех этих случаях объявляется новая **вакансия** на свободную должность. О вакансии известно, на какую должность она объявлена, **дата открытия** вакансии. Кроме того, предусмотрена **дата закрытия** вакансии, которая соответствует дате принятия на работу нового сотрудника на соответствующую должность.

Набор сотрудников на имеющиеся вакансии проходит на конкурсной основе. Кандидат обязан пройти собеседование.

Собеседования с кандидатами проводят сотрудники рекрутинга (отдел фирмы).

Рекрутеры являются сотрудниками фирмы. По каждой открытой вакансии рекрутеры фирмы ведут **собеседования с кандидатами**.

Для каждого **кандидата** фиксируется его **фамилия**, **место жительства**, **возраст**, **пол**. Один и тот же кандидат может проходить несколько собеседований на различные вакансии.

По каждому **собеседованию** фиксируется **дата проведения**, **код** рекрутера (равен его коду, как сотрудника фирмы) и полученная кандидатом **оценка** по 10-бальной шкале и **реальный оклад**, который обсуждался с кандидатом. Предусмотрена также **отметка** о результате. Если в итоге кандидат принят на вакансию, то для соответствующего собеседования проставляется признак «Yes», в противном случае - «No».

Получаем следующий выделенный список существительных:

- штатное расписание
- должность
- минимальный оклад
- максимальный оклад
- код подразделения
- сотрудник
- фамилия
- дата рождения
- адрес
- образование
- дата зачисления
- оклад

- дата увольнения
- вакансия
- дата открытия
- дата закрытия
- собеседование
- кандидат
- фамилия кандидата
- место жительства
- возраст
- пол
- оценка
- реальный оклад
- отметка
- дата проведения
- код

Теперь необходимо определить, какие из этих существительных будут представлять сущности, а какие – атрибуты сущностей. Представим список сущностей, а в скобках укажем атрибуты каждой из них. Среди атрибутов сущностей выделим первичный ключ и изобразим его подчеркиванием.

Выделение сущностей

1. Штатное расписание (уникальный номер должности, должность, код подразделения, минимальный оклад, максимальный оклад).
2. Сотрудник (код, фамилия, дата рождения, образование, адрес, оклад, дата зачисления, дата увольнения).
3. Вакансия (уникальный номер вакансии, дата открытия, дата закрытия).
4. Собеседование (уникальный номер собеседования, дата проведения, оценка, отметка, реальный оклад)
5. Кандидат (уникальный номер кандидата, фамилия кандидата, пол, возраст, место жительства).

Прежде чем переходить к установлению связей между сущностями, сделаем следующие предположения.

Предположения:

- на каждую должность можно открывать несколько вакансий
- каждое собеседование проводит один рекрутер
- сотрудник может занимать только одну должность

Установление связей между сущностями.

а) Устанавливаем связь «Занимает должность» между сущностями «Сотрудник» и «Штатное расписание». Это связь типа 1:M, поскольку в силу сделанного допущения каждый

экземпляр сотрудника может занимать только одну должность в штатном расписании. В то же время, одну должность в штатном расписании может занимать несколько сотрудников.

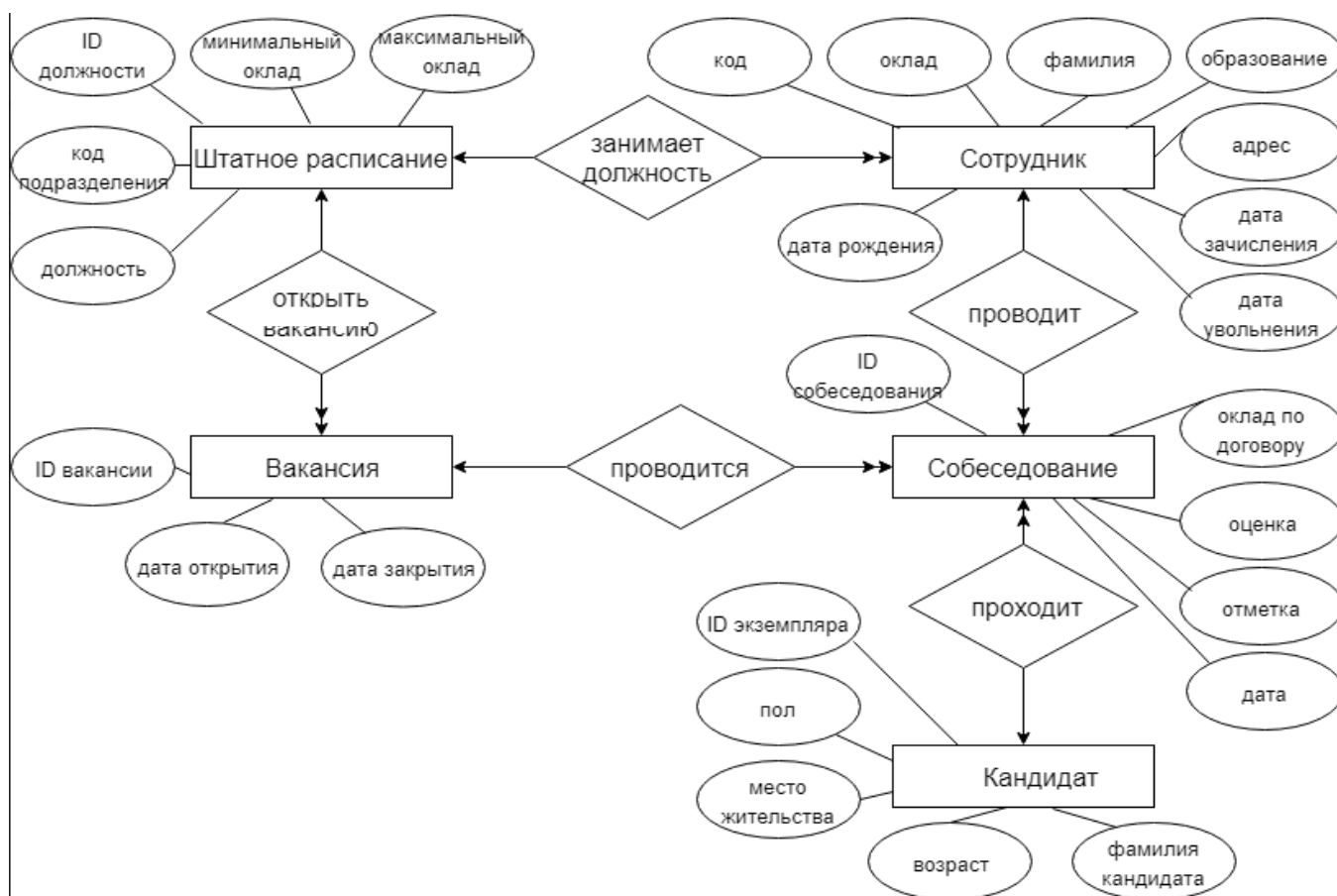
б) Устанавливаем связь «Проводит» между сущностями «Сотрудник» и «Собеседование». Эта связь типа 1:M, так как каждый экземпляр сущности «Собеседование» проводится только одним сотрудником, но каждый экземпляр сотрудника может проводить несколько собеседований.

в) Устанавливаем связь «Проходит» между сущностями «Собеседование» и «Кандидат». Это связь типа 1:M, так как каждый экземпляр собеседования проходит одним кандидатом. Но один и тот же экземпляр кандидата может проходить несколько собеседований.

г) Устанавливаем связь «Проводится» между сущностями «Собеседование» и «Вакансия». Это связь типа 1:M, поскольку каждый экземпляр собеседования проводится только на одну вакансию. Одновременно на каждую вакансию проводится несколько собеседований.

д) Устанавливаем связь «Открыть» между сущностями «Вакансия» и «Штатное расписание». Это связь типа 1:M в силу того, что на каждую должность из штатного расписания можно открыть несколько вакансий. При этом каждый экземпляр вакансии открывается лишь на одну должность из штатного расписания.

В результате получаем вариант модели предметной области.



Построение логической модели базы данных

Для построения логической модели будущей базы данных применим следующие формальные правила:

а) Каждая сущность инфологической модели отображается отдельной таблицей.

б) Каждая связь типа многие-ко-многим отображается отдельной таблицей.

в) Каждая связь типа один-ко-многим – дополнительным столбцом в дочерней таблице, и этот столбец становится внешним ключом, связанным с первичным ключом соответствующей родительской таблицы.

В результате применения этих правил получим следующий набор реляционных таблиц.

Таблица 1. Штатное расписание- Sschedule

Position_id	Position	Min_salary	Max_salary	Subdivision_code
Pk				

Таблица 2. Кандидат – Candidate

Instance_id	Gender	Can_surname	Can_address	Age
Pk				

Таблица 3. Сотрудник - Employee

Code	Surname	Birthday	Address	Education	Wage	En_date	Dis_date	Position_id
Pk								Fk

Таблица 4. Вакансия – Vacancy

Vacancy_id	Open_date	Close_date	Position_id
Pk			Fk

Таблица 5. Собеседование – Interviewing

Inter_id	Date_i	Salary	Grade	Elevation	Code	Instance_id	Vacancy_id
Pk					Fk	Fk	Fk

Разработка оперативных запросов.

1. Создать отчет о собеседованиях, проведенных сотрудником ХХХ в марте 2017 года по форме:

Дата собеседования	Id вакансии	Фамилия кандидата	Отметка «yes/no»
--------------------	-------------	-------------------	------------------

```
SELECT Date_i, Vacancy_id, Can_surname, Elevation  
FROM Employee JOIN Interviewing USING(Code) JOIN Candidate USING(Instance_id)  
WHERE (Surname="Никитина") AND (YEAR(Date_i)=2017) AND (MONTH(Date_i)=3);
```

2. Создать отчет о закрытых вакансиях, открытых в марте 2017 года по форме:

Номер вакансии	Название вакансии	Количество дней, когда вакансия была не занята (NULL для открытых)	Количество кандидатов
----------------	-------------------	---	--------------------------

```
SELECT Vacancy_id, Position, (TO_DAYS(Close_date)-TO_DAYS(Open_date)) as Day, Count(*) as Num  
FROM Vacancy JOIN Interviewing USING(Vacancy_id) JOIN Sschedule USING(Position_id)  
WHERE (YEAR(Open_date)=2017) AND (MONTH(Open_date)=3)  
GROUP BY Vacancy_id;
```

3. Показать все сведения о самом молодом сотруднике, проводившем собеседование в апреле 2017 года.

```
SELECT *  
FROM Employee  
WHERE Birthday= (SELECT MAX(Birthday)  
FROM Employee JOIN Interviewing USING(Code)  
WHERE YEAR(Date_i) = 2017 AND MONTH(Date_i)=4);
```

4. Показать все сведения о сотрудниках, никогда не проводивших собеседования (с помощью левостороннего соединения).

```
SELECT Code, Surname, Address, Birthday, Position_id, Wage, En_date  
FROM Employee LEFT JOIN Interviewing USING(Code)  
WHERE Inter_id IS NULL;
```

5. Показать сотрудников, которые не проводили собеседований в марте 2017 года.

```
SELECT Code, Surname  
FROM Employee LEFT JOIN (SELECT *  
FROM Interviewing  
WHERE YEAR(Date_i)=2017 AND MONTH(Date_i)=3)i2017  
USING(Code)  
WHERE Inter_id IS NULL;
```

6. Показать вакансию, на которую было проведено наибольшее число собеседований с кандидатами (с помощью view).

```
CREATE VIEW MAXVAC(Vacancy_id, Count_inter) AS  
SELECT Vacancy_id, COUNT(*)  
FROM Interviewing  
GROUP BY Vacancy_id;
```

```
SELECT Vacancy_id, MAX(Count_inter) as Num  
FROM MAXVAC;
```

Разработка стандартного статистического отчета

Предполагаем, что отчеты формируются для анализа количества собеседований у сотрудников в разные месяцы (только для тех, кто проводил собеседования в конкретном месяце).

В конце каждого месяца составляется отчет о проведении собеседований рекрутерами (сотрудниками фирмы) по следующей форме:

Код сотрудника	Фамилия сотрудника	Количество собеседований	Месяц	Год

Отчеты будут сохраняться в информационной системе, для чего необходимо добавить еще одну таблицу.

Разработаем процедуру, которая будет получать на вход месяц и год, а потом будет вычислять количество собеседований для каждого из сотрудников, проводивших их.

Разработка плана хранимой процедуры

1. В качестве входных параметров в процедуру должны передаваться год и месяц, для которых должны быть построены отчеты. Количество собеседований будет вычисляться для всех сотрудников, которые их проводили в заданный период.
2. Декларируется курсор, в который из соединения таблиц Employee и Interviewing помещаются код сотрудника, его фамилия и количество собеседований, который проводились в заданный период.
3. Декларируется обработчик, который будет отслеживать конец курсора.
4. Открывается курсор.
5. Открывается цикл, в котором извлекается очередная запись из курсора в локальные переменные numb, FAM, KOLV. В таблицу Otchet заносится запись с 5 значениями : 3 из курсора и 2 – из входных параметров процедуры.
6. Цикл завершается после обработки всех записей в курсоре.
7. Закрывается курсор и заканчивается процедура.

Текст хранимой процедуры

```
CREATE PROCEDURE `company`.`sobes` (inp_year integer, inp_month integer)
BEGIN
    DECLARE numb INTEGER;
    DECLARE FAM VARCHAR(24);
    DECLARE KOLV INTEGER;
    DECLARE done INTEGER DEFAULT 0;
    DECLARE C1 CURSOR FOR
        SELECT Code, Surname, COUNT(*)
        FROM Employee JOIN Interviewing USING(Code)
        WHERE (Year(Date_i)=inp_year) AND (MONTH(Date_i)=inp_month)
        GROUP BY Code;
    DECLARE EXIT HANDLER FOR SQLSTATE '02000' SET done=1;

    OPEN C1;
    WHILE done=0 DO
        FETCH C1 INTO numb, FAM, KOLV;
        INSERT OTCHET
        VALUES(NULL, numb, FAM, KOLV, inp_year, inp_month);
    END WHILE
    CLOSE C1;
END
```

Разработка приложения конечного пользователя

Запуск приложения

При запуске приложения загружается статическая HTML страница start.html. Она представляет из себя список из 6 запросов и одной процедуры. Для запросов и процедуры присутствует словесное описание, необходимые поля для ввода параметров и кнопка для их отправки контроллеру.

Контроллеры 1-6 запросов

На стартовой странице start.html клиент должен ввести в нужном запросе необходимые ему параметры и нажать на кнопку их отправки. При отправке параметров форма передает управление контроллеру необходимого запроса.

Контроллеры выполняют ряд действий:

- Подключаются к базе данных
- Формируют текст соответствующего запроса и передают его на выполнение SQL серверу.
- Переносят результаты запроса в двумерный ассоциированный массив.
- Построчно переносят ассоциированный массив в результирующую страницу и отсылают ее пользователю. Все сформированные таким образом страницы имеют ссылку для возврата в главное меню

На экране появляется таблица с необходимыми данными.

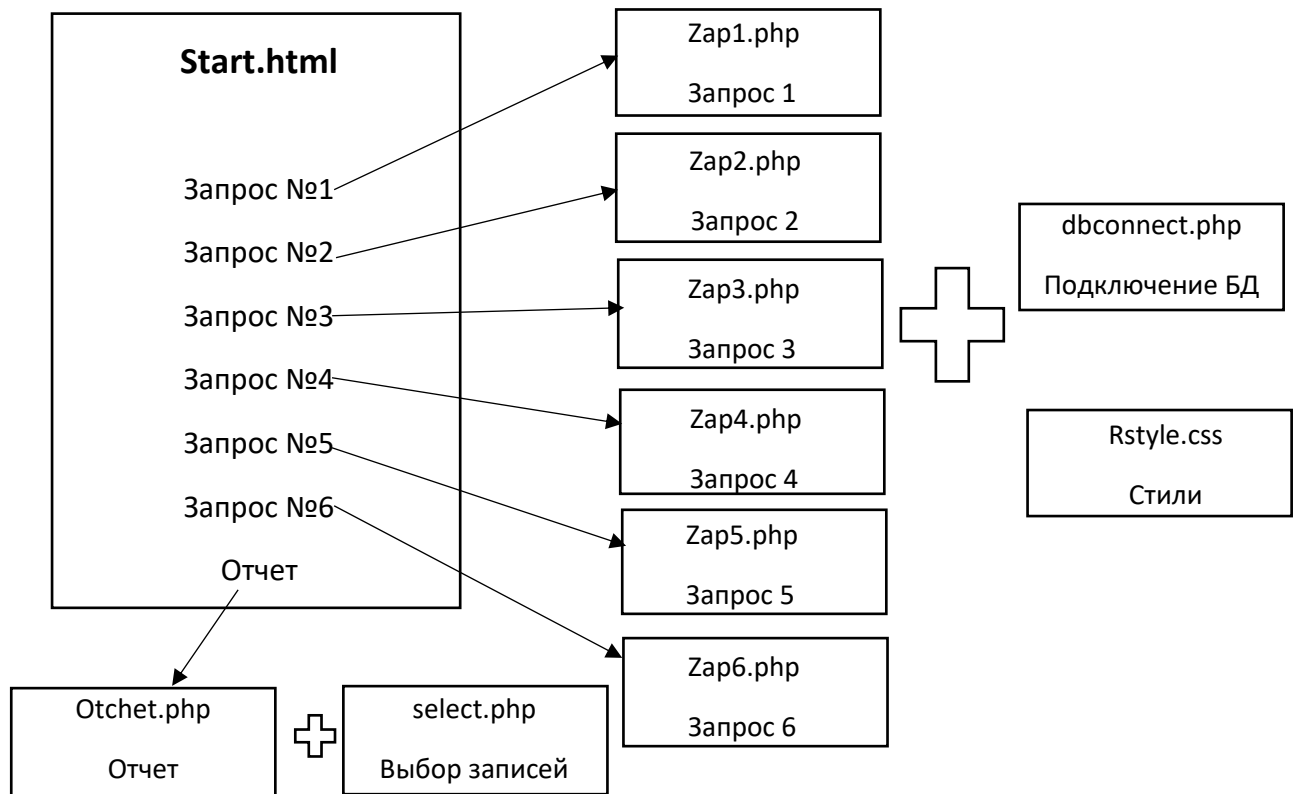
Контроллер процедуры

Контроллер должен запустить отложенную ранее процедуру для создания в базе данных новых записей в таблице Otchet. При запуске хранимой процедуры ей передаются входные параметры – месяц и год. Они получаются на статической HTML странице start.html. Форма ввода с этой страницы содержит все необходимые поля для ввода параметров и кнопку для отправки их контроллеру создания отчета. При отправке параметров форма передает управление контроллеру создания отчета.

Контроллер создания отчетов выполняет следующие действия:

- Соединяется с базой данных.
- Проверяет, не создавались ранее отчеты с полученными от пользователя параметрами.
- Если отчеты ранее не создавались, то запускает хранимую процедуру создания отчетов и передает в нее полученные от пользователя параметры.
- По окончании работы процедуры передает пользователю сообщение об успешном создании отчетов и ссылкой для возврата в главное меню.
- Если запрошенный для создания отчет уже создавался ранее, то контроллер сообщает об этом пользователю и выводит отчет на экран.

Структура приложения



Заключение

Была сформирована база данных для фирмы. Она способна осуществлять 6 оперативных запросов и запускать процедуру. Был разработан интерфейс конечного пользователя для удобства взаимодействия с базой. Все поставленные цели были выполнены.