МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ

ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»



Основная профессиональная образовательная программа

Направление подготовки 09.03.01 Информатика и вычислительная техника

Направленность (профиль) «Технологии разработки программного обеспечения»

форма обучения – очная

**ОТЧЁТ**

по реализации проекта для дисциплины «Базы данных»  
по направлению “09.03.01 – Информатика и вычислительная техника”  
(профиль: “Технологии разработки программного обеспечения ”)

Преподаватель: к.ф-м.н., доцент кафедры ИТиЭО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Жуков Н. Н.)

Преподаватель: ассистент кафедры ИТиЭО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Иванова Е. А.)

Студенты 2 курса:

Иванов И. И. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Смирнов И. И. \_\_\_\_\_\_\_\_\_\_\_\_

Петров И. И. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург  
20\_\_

Оглавление

[Ответственные 2](#_Toc515619189)

[Предметная область 2](#_Toc515619190)

[Ход выполнения нормализации 2](#_Toc515619191)

[Объяснение выбранной СУБД 3](#_Toc515619192)

[ER – диаграмма 3](#_Toc515619193)

[Исходных текст запросов 3](#_Toc515619194)

[Таблицы 3](#_Toc515619195)

[Индексы 3](#_Toc515619196)

[Триггеры 3](#_Toc515619197)

[Процедуры 3](#_Toc515619198)

[Функции 3](#_Toc515619199)

# Ответственные

В этом разделе необходимо отразить ответственных за данный проект. Описать конкретно: кто и какую часть проекта выполнял.

**Пример:** Иванов И.И. – разработчик проекта. В обязанности Иванова И.И. входил процесс нормализации базы данных. При выполнении данного задания были использованы знания по следующим формам нормализации: 1НФ-3НФ.

….

# Предметная область

В этом разделе необходимо прописать предметную область, по которой выполнялся проект.

**Пример:** Магазин, являющийся заказчиком по разработке базы данных, занимается продажей спортивных товаров. Команде разработчиков были переданы данные в виде электронных таблиц, отражающие состояние продаж спортивных товаров.

По мере увеличения объема данных, потребовалось организовать их в виде базы данных:

* спроектировать реляционную базу данных (был выявлен список сущностей, список атрибутов сущностей и проанализированы взаимосвязи между сущностями);
* провести нормализацию полученных сущностей;
* представить ER-диаграмму;
* …..

…

# Ход выполнения нормализации

В этом разделе необходимо прописать шаги проведения нормализации. Если есть необычные свойства атрибутов или была выполнена сложная декомпозиция сущностей - опишите эти моменты в данном разделе.

Данный раздел должен полностью разъяснять ниже представленную ER-диаграмму. Основываясь на данный раздел, преподаватель должен досконально разобраться в Вашем проекте, без указаний и комментариев со стороны студента.

**Пример:** После выделения конкретных сущностей, был сформирован список атрибутов каждой сущности:

ПОКУПАТЕЛЬ:

каждый покупатель является физическим лицом и имеет: фамилию, адрес (внешний ключ) и банковские реквизиты. В качестве ключевого атрибута был использован суррогатный ключ – ID\_pokupatel. Все атрибуты данной сущности обязательны к заполнению. Атрибут «адрес» является внешним ключом к сущности АДРЕС. Командой разработчиков было принято решение вынести отдельно данную сущность с целью ….

ТОВАР:

каждый товар имеет наименование, цену, а также характеризуется единицами измерения. В качестве ключевого атрибута был использован суррогатный ключ – ID\_tovar. Все атрибуты обязательны к заполнению.

….

# Объяснение выбранной СУБД

В этом разделе необходимо объективно разъяснить выбор СУБД, которая была использована в реализации проекта командой разработчиков.

# ER – диаграмма

В этом разделе представляется ER-диаграмма, разработанная по описанию предметной области.

Будет плюсом (пока что в карму), если диаграмма будет соответствовать нотации UML или стандарту IDEF1x.

# Исходный текст запросов

## По созданию таблиц

**Пример:**

--Simple CREATE TABLE Syntax (common if not using options)

CREATE TABLE

[ database\_name . [ schema\_name ] . | schema\_name . ] table\_name

( { <column\_definition> } [ ,...n ] )

[ ; ]

## По созданию индексов

**Пример:**

--Create a clustered index on a table and use a 3-part name for the table

CREATE CLUSTERED INDEX i1 ON d1.s1.t1 (col1);

## По созданию триггеров

**Пример:**

-- SQL Server Syntax

-- Trigger on an INSERT, UPDATE, or DELETE statement to a table or view (DML Trigger)

CREATE [ OR ALTER ] TRIGGER [ schema\_name . ]trigger\_name

ON { table | view }

[ WITH <dml\_trigger\_option> [ ,...n ] ]

{ FOR | AFTER | INSTEAD OF }

{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }

[ WITH APPEND ]

[ NOT FOR REPLICATION ]

AS { sql\_statement [ ; ] [ ,...n ] | EXTERNAL NAME <method specifier [ ; ] > }

<dml\_trigger\_option> ::=

[ ENCRYPTION ]

[ EXECUTE AS Clause ]

<method\_specifier> ::=

assembly\_name.class\_name.method\_name

## По созданию процедур

**Пример:**

-- Transact-SQL Syntax for Stored Procedures in SQL Server and Azure SQL Database

CREATE [ OR ALTER ] { PROC | PROCEDURE }

[schema\_name.] procedure\_name [ ; number ]

[ { @parameter [ type\_schema\_name. ] data\_type }

[ VARYING ] [ = default ] [ OUT | OUTPUT | [READONLY]

] [ ,...n ]

[ WITH <procedure\_option> [ ,...n ] ]

[ FOR REPLICATION ]

AS { [ BEGIN ] sql\_statement [;] [ ...n ] [ END ] }

[;]

<procedure\_option> ::=

[ ENCRYPTION ]

[ RECOMPILE ]

[ EXECUTE AS Clause ]

## По созданию функций

**Пример:**

-- Transact-SQL Scalar Function Syntax

CREATE [ OR ALTER ] FUNCTION [ schema\_name. ] function\_name

( [ { @parameter\_name [ AS ][ type\_schema\_name. ] parameter\_data\_type

[ = default ] [ READONLY ] }

[ ,...n ]

]

)

RETURNS return\_data\_type

[ WITH <function\_option> [ ,...n ] ]

[ AS ]

BEGIN

function\_body

RETURN scalar\_expression

END

[ ; ]