

**МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ  
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)  
Факультет инноваций и высоких технологий  
Кафедра «Алгоритмы и технологии программирования»**

**Курсовая работа по курсу «Базы данных»**

Задача курсовой работы состоит в получении практических навыков использования промышленных СУБД. Выбор конкретной СУБД – на усмотрение студента. Основное условие – СУБД должна быть реляционной. Рекомендуется (читай – обязательно) использовать PostgreSQL.

**Задание:**

1. Выбрать предметную область для моделирования, согласовать ее с семинаристом. В области должно быть 4-6 основных сущностей.
2. Спроектировать концептуальную, логическую и физическую модели базы данных. Проектировать можно (и нужно) в редакторе Gliffy (<https://www.gliffy.com/>).
  - a. Концептуальная модель должна описывать все сущности, входящие в моделируемую область, без уточнения атрибутов и декомпозиции. **4-6 основных сущностей.**  
*Подсказка:* рисунок с квадратами Склад-Товар и связью N-N между ними.
  - b. Логическая модель должна описывать таблицы, из которых состоит проектируемая база данных и атрибуты. Логическую модель требуется проектировать с учетом возможной декомпозиции таблиц. Проектируемая база должна находиться во 2НФ или 3НФ с обоснованием выбора НФ. Допустимо использование современных подходов к моделированию – DataVault, Anchor и других, по согласованию с семинаристом. **После нормализации ~10 сущностей (для DataVault ~ 15, для Anchor ~60-70).**  
*Подсказка:* ER-диаграмма с сущностями и атрибутами.
  - c. Физическая модель должна описывать, каким образом спроектированная база хранится в СУБД. Требуется описание каждого поля каждой таблицы: название и описание поля, тип данных, ограничения (constraint) наложенные на поле. **По одной таблице на каждую сущность.** Приблизительный формат для описания каждой таблицы:

POLICY_COVERAGE / Покрытие полиса			
Название	Описание	Тип данных	Ограничение
policy_coverage_cd	Трёхбуквенный код покрытия по полису	VARCHAR(3)	NOT NULL PRIMARY KEY
policy_coverage_ru_nm	Русское наименование покрытия	VARCHAR(100)	
policy_coverage_en_nm	Английское наименование покрытия	VARCHAR(100)	
insurance_type_code	Тип страхования	VARCHAR(100)	
insurance_kind_code	Код типа страхования для системы "Диасофт"	VARCHAR(15)	
accounting_group_no	Учётная группа, по которой согласно ФЗ формируются страховые резервы	INTEGER	
processed_dttm	Дата\время появления записи в словаре	TIMESTAMP(0)	

При желании можно добавить столбцы PK, FK. Ключи уже указаны в логической модели данных, но для удобства чтения спецификаций на каждую таблицу их можно добавить.

*Подсказка:* по таблице на каждую сущность из ER-диаграммы.

3. Написать DDL-скрипты для создания пустой БД.

4. Заполнить БД данными (~100 записей в сумме, по 5-10 записей в каждую таблицу). Для заполнения можно использовать как рукописные запросы (INSERT INTO.. SELECT UNION SELECT), так и внешние источники данных (XLS, CSV). На усмотрение студента.
5. Сформулировать **5 смысловых запросов** к БД. Написать их на SQL.
6. Написать CRUD-запрос (*подсказка*: 4 запроса) к **двум любым** таблице БД.
7. Создать по 1 представлению на каждую таблицу. В представлениях должен быть реализован механизм маскирования личных (секретных) данных и скрывания технических полей (суррогатных ключей и т.п.)  
*Подсказка*: Список “секретных” атрибутов определяется студентом. Если, например, номер банковской карты пользователя – секретный, вместо реального номера при запросе из представления должна выводиться заглушка, например ‘555947\*\*\*\*\*8424’. При этом хранимые данные не должны меняться.
8. Создать 2 сложных представления (**с джойном**).
9. \*Создать 2 триггера на любые таблицы БД. Логика работы обговаривается с семинаристом.
10. \*Создать хранимую процедуру. Логика работы обговаривается с семинаристом.
11. \*\*Пообщаться с семинаристом при сдаче. Быть готовым написать какие-нибудь запросы руками, что-нибудь показать или объяснить.  
*Подсказка*: семинарист добрый, если все сделано самостоятельно.

### Вес заданий:

Задание	Комментарий	Балл
1	Необходимое условие сдачи зачета	-
2	Необходимое условие сдачи зачета	-
3	Для сдачи продемонстрировать работу всех скриптов	1
4	Если заполняется запросом – продемонстрировать запрос. Если из файла – залить новые данные из файла.	1
5	По 0,5 за каждый запрос. «Смысловость» запроса предварительно лучше согласовать с семинаристом.	2,5
6	Для сдачи продемонстрировать работу 4-х запросов SELECT, INSERT, UPDATE, DELETE.	0,5
7	Для сдачи продемонстрировать скрипты создания представлений и селекты из этих представлений.	1
8	Для сдачи каждое представление должно реализовывать некую логику и иметь смысл.	1
9	Обговаривается с семинаристом.	~1,5 (Отл(8) за 1 выполненную звездочку, отл(10) за 2)
10	Обговаривается с семинаристом.	~1,5 (Отл(8) за 1 выполненную звездочку, отл(10) за 2)
11	Супер необходимое условие сдачи зачета.	Понижающе-повышающий коэффициент за качество ответа.

**Важно!** Итоговый балл (из 10) переводится в 3-балльную систему умножением на 3/10.

## Требования к наименованию объектов:

### Общие правила

1. В названии объектов следует использовать английские слова в единственном числе. Для перевода русских определений на английский полезно использовать онлайн-сервис [multitran](#).
2. Для обозначения одного понятия в именах разных объектов следует использовать один и тот же термин.
3. Имя объекта не должно превышать 32 символа, считая разделители.

### Имена таблиц

1. Название таблицы должно содержать название описываемой сущности.
2. В имени таблицы на первом месте всегда идет описываемая сущность. Любые суффиксы должны добавляться после названия сущности.  
*Пример:* POLICY\_CASCO\_OPTIONS, CUSTOMER\_DOCUMENT\_TYPE.
3. Если таблица является таблицей связи, то её имя должно состоять из имён связываемых сущностей, разделённых символами ‘\_X\_’  
*Пример:* CUSTOMER\_X\_AGREEMENT
4. Если полное имя таблицы не уместается в ограничение по длине, сокращать его нужно начиная с конца.

### Имена полей

1. Имя поля должно содержать окончание из числа представленных далее:

Окончание	Семантика	Множество значений	Примеры
<b>Идентификаторы</b>			
_id	Идентификатор сущности	Целые числа, строки	activity_id, service_request_id, nice_evaluation_id
_no	Номерной идентификатор	Натуральные числа, строки	mobile_phone_no, passport_no, social_security_no
_series	Серия (как часть номерного идентификатора)	Натуральные числа, строки	passport_series, license_series
<b>Классификаторы</b>			
_dk	Постоянный ключ элемента словаря	Натуральные числа	policy_product_dk
_cd	Трёхсимвольный словарный код	Строки верхнего регистра длиной в три символа	account_type_cd, currency_cd, source_system_cd
_code	Бизнес-код	Натуральные числа, строки	error_code, agent_comp_code, bki_code
<b>Дата и время</b>			
_dt	Дата	Дата в формате DD-MM-YYYY	activation_dt, birth_dt
_dtm	Дата и время	Дата и время в формате DD-MM-YYYY HH24:MI:SS	transaction_dtm, valid_from_dtm, utilization_dtm
_sec	Величина временного интервала в секундах	Натуральное число +0	duration_sec
_min	Величина временного интервала в минутах	Натуральное число +0	
_ms	Величина временного интервала в миллисекундах	Натуральное число +0	

<b>Числительные</b>			
_amt	Сумма (остаток)	Вещественные числа	transaction_amt, balance_amt, payment_amt
_num	Порядковый номер	Натуральные числа	bank_card_reopen_num,
_coeff	Коэффициент	Вещественные числа	
_cnt	Количество экземпляров	Натуральные числа + 0	
_prt	Доля	Вещественные числа от 0 до 1	
_rate	Ставка/Отношение	Вещественные числа	currency_rate
_pct	Проценты	Вещественные числа от 0 до 100	psk_normal_pct, purch_int_pct
_flg	Бинарный флаг	0/1	deleted_flg, success_flg, actual_flg
<b>Текстовые</b>			
_nm	Имя	Строка	file_nm, loyalty_program_type_nm, middle_nm
_desc	Описание	Строка	add_phone_desc, mdm_party_match_desc
_url	Адрес страницы в Internet	Строка в формате 'http[s]://%'	merchant_url, success_url
_txt	Текст в свободной форме	Строка	comment_txt, error_message_txt
_value	Значение параметра; чаще всего - в структурах формата key -value.	Строки, реже числа	param_value, criterion_value, element_value
_title	Заголовок	Строка	job_title

- Исключением являются общепринятые аббревиатуры – их можно использовать в качестве названий полей без каких-либо добавок - как есть.