

Практическое занятие № 15

1. Наименование практического занятия: составление программ для работы с базами данных в IDE PyCharm Community.
2. Количество часов: 8
3. Место проведения: главный корпус РКСИ, ауд. 420.
4. Цели практического занятия: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ для работы с базами данных в IDE PyCharm Community.

Критерии оценивания:

Для всех оценок:

- Наличие всех таблиц в БД и связей между ними посредством внешних ключей.
- Корректность создания полей в каждой таблице с учетом их типа и ограничений.
- Корректность реализации SQL-запросов на выборку данных, т.е. запросы выполняются и выводят корректные результаты, но могут быть неоптимальными или неэффективными.
- Наличие запросов на обновление данных в БД, т.е. выполнение запросов происходит без ошибок и обновление данных происходит корректно.
- Отсутствие нарушений целостности данных, т.е. удаление или обновление записей в таблицах не приводит к ошибкам или нарушениям связей между таблицами.
- Корректность использования внешних ключей в таблицах "Бронирования" и "Туры" для связи с таблицами "Туристы".
- Правильность выбора типов данных для полей таблиц, а также корректность выбора ограничений на поля.
- Умение разобраться в структуре БД и понимание основных принципов работы с SQL-запросами.
- Соответствие кода базы данных и запросов стандартам написания кода.
- Код программы предоставлен в установленный срок.
- Код программы не должен содержать синтаксических и логических ошибок, содержать комментарии, соответствовать PEP 8.

Оценка "5":

- Создана БД со всеми таблицами, связанными между собой внешними ключами
- Все таблицы содержат необходимые поля, соответствующие требованиям задания
- Созданы все SQL-запросы на выборку данных из БД, все запросы корректно работают и дают верные результаты
- Реализованы все SQL-запросы на обновление данных в БД, все запросы корректно работают и изменяют нужные поля

Оценка "4":

- Создана БД с тремя таблицами, связанными между собой внешними ключами
- Все таблицы содержат необходимые поля, соответствующие требованиям задания
- Созданы SQL-запросы на выборку данных из БД, все запросы корректно работают и дают верные результаты
- Реализовано не менее половины всех SQL-запросы на обновление данных в БД или не менее половины всех SQL-запросов корректно работают и/или не все изменения соответствуют требованиям задания.

Оценка "3":

- Создана БД с тремя таблицами, связанными между собой внешними ключами
- Все таблицы содержат необходимые поля, соответствующие требованиям задания
- Созданы SQL-запросы на выборку данных из БД, все запросы корректно работают и дают верные результаты
- Реализовано менее половины всех SQL-запросы на обновление данных в БД или менее половины всех SQL-запросов корректно работают и/или не все изменения соответствуют требованиям задания.

Варианты заданий

Вариант 1.

Разработать БД «Турист» с тремя таблицами, установить связь между таблицами. Заполнить таблицы произвольными данными (не менее 10 записей). Реализовать SQL-запросы на выборку, обновление, удаление данных из БД.

Таблица "Туристы"

id (INT, PK) - уникальный идентификатор туриста
имя (VARCHAR)
фамилия (VARCHAR)
пол (VARCHAR)
дата_рождения (DATE)
номер_телефона (VARCHAR)
электронная_почта (VARCHAR)
Таблица "Туры"

id (INT, PK) - уникальный идентификатор тура
название (VARCHAR)
страна (VARCHAR)
город (VARCHAR)
дата_начала (DATE)
дата_окончания (DATE)
цена (DECIMAL)
Таблица "Бронирования"

id (INT, PK) - уникальный идентификатор бронирования
id_туриста (INT, FK) - идентификатор туриста, который сделал бронирование
id_тура (INT, FK) - идентификатор тура, на который было сделано бронирование
дата_бронирования (DATE)
кол-во_туристов (INT)

Для установки связей между таблицами использовать внешние ключи (FK) на таблицах "Бронирования" и "Туры", которые ссылаются на соответствующие поля таблицы "Туристы" и "Туры".

SQL-запросы на выборку данных из БД:

1. Вывести список всех туристов
2. Вывести список всех туров, отсортированных по цене в порядке убывания
3. Вывести список всех бронирований, сделанных в заданном городе
4. Вывести список всех туристов, сделавших бронирование в определенный период времени
5. Вывести список всех туров с указанием названия страны и города
6. Вывести список всех туристов, женщин, у которых дата рождения позже 01.01.1990
7. Вывести список всех туров, цена которых больше 5000
8. Вывести список всех туристов, которые сделали бронирование на конкретный тур
9. Вывести список всех туристов, которые сделали бронирование на тур в указанную дату

10. Вывести список всех туристов, у которых номер телефона начинается на "+7"

SQL-запросы на обновление данных из БД:

1. Изменить дату начала тура с id=1 на '2023-05-01':
2. Обновить цену тура с id=7 на 1500:
3. Изменить номер телефона туриста с id=5 на '+1 (555) 123-4567':
4. Изменить дату бронирования с id=3 на '2023-04-05':
5. Обновить количество туристов в бронировании с id=8 на 3:
6. Изменить дату окончания тура с id=2 на '2023-08-31':
7. Обновить электронную почту туриста с id=1 на 'new_email@example.com':
8. Изменить дату начала тура с id=4 на '2023-06-15':
9. Обновить дату начала тура на 2023-05-01 для всех туров, где страна = 'Испания':
10. Обновление цены на тур "Греция-отдых на море" на 1500 у.е.
11. Обновление даты начала тура "Испания-путешествие по городам" на 2023-06-01.
12. Обновление количества туристов в бронировании с id 1002 на 3 человека.
13. Обновление номера телефона у туриста с id 2001 на +1 (123) 456-7890.
14. Обновление даты начала тура на 2024-07-01 для всех туров, цена которых меньше 2000 у.е.
15. Обновление электронной почты у всех туристов из России на new_email@example.com.
16. Обновление даты начала тура на 2023-08-15 для всех бронирований с количеством туристов больше 2.
17. Обновление названия тура на "Египет-отдых на курорте" для всех бронирований с id_тура равным 1003.

SQL-запросы на удаление данных из БД:

1. Удалить все бронирования, связанные с туристом с id=1
2. Удалить все бронирования, связанные с туром с id=2
3. Удалить все бронирования, сделанные в определенную дату
4. Удалить всех туристов, которые сделали бронирование на тур с id=3
5. Удалить все бронирования, сделанные туристом с определенным номером телефона
6. Удалить все бронирования, сделанные туристом с определенной электронной почтой
7. Удалить все бронирования на туры, начинающиеся после определенной даты
8. Удалить всех туристов, которые забронировали тур в определенную страну
9. Удалить все бронирования на туры, заканчивающиеся до определенной даты
10. Удалить все бронирования, сделанные на тур с определенной ценой

Вариант 2.

Разработать БД «ОПТОВАЯ БАЗА». Структура БД состоит из пяти таблиц: Товары, Магазины, Заявки магазинов, Количество товаров на складе, Состав. Установить связь между таблицами. Заполнить таблицы произвольными данными (не менее 10 записей). Реализовать SQL-запросы на выборку, обновление, удаление данных из БД.

Таблица "Товары"

id (INT, PK) - уникальный идентификатор товара
название (VARCHAR)
описание (VARCHAR)
единица измерения (VARCHAR)

Таблица "Магазины"

id (INT, PK) - уникальный идентификатор магазина
название (VARCHAR)
адрес (VARCHAR)
телефон (VARCHAR)

Таблица "Заявки магазинов"

id (INT, PK) - уникальный идентификатор заявки
id_магазина (INT, FK) - идентификатор магазина, который подал заявку
дата_заявки (DATE)

Таблица "Количество товаров на складе"

id (INT, PK) - уникальный идентификатор записи
id_товара (INT, FK) - идентификатор товара
количество (INT)

Таблица "Состав"

id (INT, PK) - уникальный идентификатор записи
id_заявки (INT, FK) - идентификатор заявки
id_товара (INT, FK) - идентификатор товара
количество (INT)

SQL-запросы на выборку данных:

1. Вывести список всех товаров и их описания:
2. Вывести список всех магазинов и их адресов:
3. Вывести список всех заявок магазинов и даты, на которые они были поданы:
4. Вывести список товаров и количество их наличия на складе:
5. Вывести список товаров и количество их наличия на складе в порядке убывания количества:
6. Вывести список всех заявок магазинов и товаров, которые были в них заказаны:
7. Вывести список всех товаров, у которых на складе количество меньше минимально допустимого:

8. Вывести список всех заявок магазинов, которые были сделаны в определенный период времени:
9. Вывести список всех магазинов, у которых суммарное количество товаров на складе меньше заданного значения:

SQL-запросы на обновление данных из БД:

1. Обновить количество товара на складе для конкретного товара
2. Обновить название товара в заявке
3. Обновить количество товара в заявке
4. Обновить адрес магазина, который подал заявку
5. Обновить дату заявки для конкретного магазина
6. Обновить количество товара на складе для нескольких товаров
7. Обновить описание товара и количество на складе для конкретного товара
8. Обновление количества товаров на складе, учитывая выполненную заявку магазина
9. Обновление количества товаров на складе, учитывая выполненную заявку магазина с учетом конкретного товара
10. Обновить название магазина, который подал заявку, и адрес магазина для конкретной заявки.
11. Обновить название магазина в заявке, которую подал конкретный магазин
12. Обновить адрес магазина и количество товара в заявке для конкретного товара
13. Обновить описание товара и количество на складе для нескольких товаров

1. SQL-запросы на удаление данных из БД:

2. Удаление заявки магазина и соответствующих записей в таблице состава
3. Удалить из таблицы "Количество товаров на складе" записи, соответствующие товарам, не имеющим заявок в таблице "Состав"
4. Удалить из таблицы "Заявки магазинов" все заявки магазинов, адрес которых начинается на "ул. Ленина"
5. Удалить из таблицы "Состав" записи, соответствующие товарам, которых нет на складе (количество = 0)
6. Удалить из таблицы "Магазины" магазины, в которых не было заявок в течение последнего месяца
7. Удалить из таблицы "Товары" товары, которые не были заказаны ни разу
8. Удалить из таблицы "Количество товаров на складе" записи, соответствующие товарам, которые не были заказаны ни разу
9. Удалить из таблицы "Состав" записи, соответствующие заявкам, которые были поданы более месяца назад

Вариант 3.

Разработать БД «ЗАПЛАТА», содержащую две таблицы *Анкета* и *Больничные листы*. Установить связь между таблицами. Заполнить таблицы произвольными данными (не менее 10 записей). Реализовать SQL-запросы на выборку, обновление, удаление данных из БД.

Таблица "Анкета"

id (INT, PK) - уникальный идентификатор сотрудника

имя (VARCHAR)

фамилия (VARCHAR)

дата_рождения (DATE)

пол (VARCHAR)

дата_найма (DATE)

должность (VARCHAR)

отдел (VARCHAR)

базовая_ставка (DECIMAL)

Таблица "Больничные листы"

id (INT, PK) - уникальный идентификатор больничного листа

id_сотрудника (INT, FK) - идентификатор сотрудника, на которого выписан больничный лист

дата_начала (DATE)

дата_окончания (DATE)

причина (VARCHAR)

диагноз (VARCHAR)

оплачен (BOOLEAN)

В данной структуре таблица "Больничные листы" связана с таблицей "Анкета" через внешний ключ id_сотрудника. Это означает, что каждый больничный лист относится к определенному сотруднику из таблицы "Анкета".

SQL-запросы на выборку данных:

1. Вывести список всех сотрудников и их должностей
2. Вывести список всех сотрудников и их базовых ставок
3. Вывести список всех сотрудников, работающих в отделе "IT"
4. Вывести список всех сотрудников, принятых на работу после 1 января 2022 года
5. Вывести список всех больничных листов, выписанных сотруднику с id = 42
6. Вывести список всех больничных листов, оплаченных компанией
7. Вывести список всех сотрудников, имеющих больничные листы на текущий месяц
8. Вывести среднюю базовую ставку всех сотрудников
9. Вывести список всех сотрудников, имеющих базовую ставку выше 100 000
10. Вывести список всех сотрудников и общее количество дней, проведенных ими на больничном
11. Вывести информацию о сотрудниках и их больничных листах за последний месяц
12. Вывести среднюю продолжительность больничных листов сотрудников в каждом отделе
13. Вывести список сотрудников и информацию о последнем больничном листе, который они оформляли
14. Вывести список сотрудников и информацию о первом больничном листе, который они оформляли

15. Вывести список сотрудников и суммарную продолжительность их больничных листов в текущем году

SQL-запросы на обновление данных из БД:

1. Обновить базовую ставку сотрудника на определенной должности.
2. Обновить отдел для всех сотрудников в определенном диапазоне возраста.
3. Обновить дату найма для сотрудника, получившего повышение.
4. Обновить причину больничного листа для сотрудника.
5. Обновить базовую ставку сотрудника в таблице "Анкета" на определенный процент, используя INNER JOIN с таблицей "Больничные листы". При этом необходимо исключить из обновления сотрудников, у которых были неоплаченные больничные листы.
6. Обновить дату начала больничного листа в таблице "Больничные листы" на определенную дату, используя INNER JOIN с таблицей "Анкета". При этом необходимо исключить из обновления больничные листы с уже пройденной датой начала
7. Обновить причину больничного листа в таблице "Больничные листы" на определенное значение для всех сотрудников, работающих в отделе "Бухгалтерия".

SQL-запросы на удаление данных из БД:

1. Удалить все записи о больничных листах для сотрудника с именем "Иван"
2. Удалить все записи о больничных листах для сотрудника с фамилией "Петров"
3. Удалить все записи о больничных листах для сотрудника с должностью "Менеджер"
4. Удалить все записи о больничных листах для сотрудника с отделом "Отдел продаж"
5. Удалить все записи о больничных листах для сотрудника женского пола
6. Удалить все записи о больничных листах для сотрудников старше 50 лет
7. Удалить все записи о неоплаченных больничных листах
8. Удалить все записи о больничных листах, дата окончания которых прошла
9. Удалить все записи о больничных листах, начиная с определенной даты
10. Удалить все записи о больничных листах, закончившихся до определенной даты
11. Удалить все больничные листы сотрудника с именем "Иван" из таблицы "Больничные листы"
12. Удалить все больничные листы сотрудников, чьи фамилии начинаются на букву "С" из таблицы "Больничные листы"
13. Удалить все больничные листы, которые еще не были оплачены, у сотрудников с должностью "Менеджер" из таблицы "Больничные листы"
14. Удалить все больничные листы, выписанные сотрудникам отдела "IT" в период с 1 января
15. Удалить все больничные листы, связанные со сотрудниками старше 50 лет из таблицы "Больничные листы"

Вариант 4 (можно 4 человека)

Разработать БД ДЕКАНАТ содержащую следующие таблицы. Установить связь между таблицами. Реализовать SQL-запросы на выборку данных из БД. Заполнить таблицы произвольными данными (не менее 10 записей). Реализовать SQL-запросы на выборку, обновление, удаление данных из БД.

Структура БД "Деканат":

Таблица "Факультеты":

id (INT, PK) - уникальный идентификатор факультета
название (VARCHAR)

Таблица "Кафедры":

id (INT, PK) - уникальный идентификатор кафедры
название (VARCHAR)
id_факультета (INT, FK) - идентификатор факультета, к которому относится кафедра

Таблица "Специальности":

id (INT, PK) - уникальный идентификатор специальности
название (VARCHAR)
id_кафедры (INT, FK) - идентификатор кафедры, которая готовит по данной специальности

Таблица "Предметы":

id (INT, PK) - уникальный идентификатор предмета
название (VARCHAR)

Таблица "Форма сдачи предмета":

id (INT, PK) - уникальный идентификатор формы сдачи предмета
название (VARCHAR)

Таблица "Учебный план":

id (INT, PK) - уникальный идентификатор учебного плана
id_специальности (INT, FK) - идентификатор специальности, для которой составлен учебный план
id_предмета (INT, FK) - идентификатор предмета, входящего в учебный план
id_формы_сдачи_предмета (INT, FK) - идентификатор формы сдачи предмета
кол_во_лекционных_часов (INT)
кол_во_практических_часов (INT)
кол_во_лаборторных_часов (INT)
курсовая_работа(bool)

Таблица "Абитуриенты":

ID	Целое число	Уникальный идентификатор абитуриента
Фамилия	Текст	Фамилия абитуриента
Имя	Текст	Имя абитуриента
Отчество	Текст	Отчество абитуриента
Пол	Текст	Пол абитуриента
Дата рождения	Дата/время	Дата рождения абитуриента
Адрес	Текст	Адрес абитуриента
Телефон	Текст	Телефон абитуриента
Email	Текст	Email абитуриента
Дата поступления	Дата/время	Дата поступления абитуриента
Специальность	Текст	Специальность, на которую зачислен абитуриент

Структура таблицы "Учебная карточка" в БД Деканат:

ID_Карточки (целое число, первичный ключ)
 ФИО студента (текст)
 Группа (текст)
 Специальность (текст)
 Предмет (текст, внешний ключ на таблицу "Предметы")
 Форма сдачи предмета (текст, внешний ключ на таблицу "Форма сдачи предмета")
 Оценка (целое число)

SQL-запросы на выборку данных из БД:

1. Вывести список всех студентов, зачисленных на факультет, с указанием номера группы.
2. Вывести список всех специальностей факультета и количество студентов, обучающихся по каждой из них.
3. Вывести список всех кафедр факультета и количество студентов, обучающихся на каждой кафедре.
4. Вывести список всех предметов и количество часов, выделенных на каждый предмет в учебном плане каждой специальности.
5. Вывести список всех студентов, у которых есть неудовлетворительные оценки (меньше 4) по любому предмету.
6. Вывести список всех предметов, которые изучают студенты первого курса.
7. Вывести список всех студентов, которые сдают курсовую работу в этом семестре.
8. Вывести список всех абитуриентов, зачисленных на специальность "Информатика и вычислительная техника".
9. Вывести список всех предметов, которые изучают студенты группы 101.
10. Вывести список студентов и их оценки за все предметы на специальности "Программная инженерия"

SQL-запросы на обновление данных из БД:

1. Обновление названия факультета с id=1 на "Новый факультет"
2. Обновление названия кафедры с id=2 на "Новая кафедра"
3. Обновление названия специальности с id=3 на "Новая специальность"
4. Обновление названия предмета с id=4 на "Новый предмет"
5. Обновление названия формы сдачи предмета с id=5 на "Новая форма сдачи"
6. Обновление количества лекционных часов на 30 для учебного плана с id=6

7. Обновление количества лекционных часов у предмета "Математика" на учебном плане специальности "Математика и информатика"
8. Обновление количества лабораторных часов и формы сдачи предмета у специальности "Программная инженерия" для предмета "Программирование"
9. Обновление количества лекционных и практических часов у предмета "Информатика" на учебном плане кафедры "Информационных технологий"
10. Обновить кол-во лекционных часов для всех предметов, где количество лекционных часов больше 30
11. Обновить фамилию и имя абитуриента по его идентификатору
12. Обновить название кафедры для всех специальностей, где кафедра имеет id = 1
13. Обновить оценку по определенному предмету и форме сдачи для конкретного студента
14. Обновить название специальности для всех студентов, где специальность имеет id = 2
15. Обновить все оценки студента с именем "Иван" на предмете "Математика" на значение 5
16. Обновить название факультета на "Факультет информационных технологий" для всех кафедр, относящихся к этому факультету
17. Обновить количество лабораторных часов на предмете "Физика" для специальности "Физика и информатика" на 30

SQL-запросы на удаление данных из БД:

1. Удалить всех абитуриентов, у которых дата поступления ранее 2020 года.
2. Удалить все учебные планы, которые не связаны ни с одной специальностью.
3. Удалить все кафедры, которые не относятся ни к одному факультету.
4. Удалить все предметы, которые не входят ни в один учебный план.
5. Удалить всех студентов, не имеющих оценок.
6. Удалить все записи из таблицы "Учебная карточка" для предмета "Математика" с формой сдачи "Экзамен"
7. Удалить все записи из таблицы "Учебная карточка" для студентов, обучающихся на факультете "Информационных технологий"
8. Удалить все записи из таблицы "Учебная карточка" для студентов, не сдавших ни одного экзамена
9. Удалить всех абитуриентов, которые не поступили на заданную специальность
10. Удалить все учебные карточки, связанные с предметом, который больше не входит в учебный план по заданной специальности
11. Удалить всех абитуриентов, не сдавших заданный предмет
12. Удалить все учебные карточки, связанные с заданным преподавателем
13. Удалить все учебные карточки, связанные с заданной формой сдачи предмета
14. Удалить всех абитуриентов, не зачисленных на заданную специальность
15. Удалить всех абитуриентов, которые подавали заявку на специальность с названием "Информатика", вместе с их учебными карточками
16. Удалить всех студентов, которые учатся на специальности "Информационные технологии" и сдали экзамен по предмету "Алгоритмы и структуры данных" на более низкую оценку, вместе с их учебными карточками
17. Удалить всех студентов, которые не сдали экзамен по предмету "Физика" в установленный срок, вместе с их учебными карточками

Вариант 5

Разработать БД Библиотека содержащую следующие таблицы. Установить связь между таблицами. Заполнить таблицы произвольными данными (не менее 10 записей). Реализовать SQL-запросы на выборку, обновление, удаление данных из БД.

Таблица Авторы:

КодАвтора (ключевое поле)

Фамилия

Имя

Таблица Книги:

КодКниги (ключевое поле)

Название

Раздел (поле подстановки из таблицы Разделы)

Издательство (поле подстановки из таблицы Издательства)

ГодИздания

МестоХранения

Таблица Разделы:

Раздел (ключевое поле)

Таблица Издательства:

Издательство (уникальный ключ)

Город

Таблица АвторКниги:

КодАвтораКниги (ключевое поле)

КодКниги

КодАвтора

SQL-запросы на выборку данных из БД:

1. Получить список всех книг, отсортированных по году издания.
2. Получить список всех книг заданного автора.
3. Получить список всех книг из заданного раздела.
4. Получить список всех книг, изданных заданным издательством.
5. Получить список всех авторов в алфавитном порядке.
6. Получить список всех книг, отсортированных по названию и году издания.
7. Получить список всех книг заданного автора, отсортированных по году издания.
8. Получить список всех книг, опубликованных в заданном году.
9. Получить список всех авторов, написавших книги для заданного издательства.
10. Получить список всех книг, в названии которых есть заданное слово.

SQL-запросы на обновление данных из БД:

1. Обновить год издания всех книг, написанных автором с фамилией "Иванов", установив год издания равным 2022:

2. Обновить название и год издания книги, хранящейся в городе "Москва", установив название "Новая книга" и год издания равным 2023:
3. Обновить название и раздел всех книг, написанных автором с именем "Александр" и фамилией "Петров", установив название "Новое название" и раздел "Фантастика":
4. Обновить название всех книг, которые были опубликованы в годы с 2010 по 2015 включительно, установив название "Старое название":
5. Обновить место хранения всех книг, написанных автором с кодом 7, установив место хранения "Библиотека №2":
6. Обновление города из таблицы Издательства по коду города в таблице Книги:
7. Обновление кода автора в таблице АвторКниги по коду автора в таблице Авторы:
8. Обновление названия раздела в таблице Книги по названию раздела в таблице Разделы:
9. Обновление года издания в таблице Книги по году издания в таблице АвторКниги:
10. Обновление места хранения в таблице Книги по названию издательства в таблице Издательства:
11. Обновление фамилии автора в таблице Авторы по коду автора в таблице АвторКниги:
12. Обновить год издания всех книг, изданных в городе "Москва", на 2022 год.
13. Обновить место хранения всех книг, написанных автором с фамилией "Иванов", на "Книжный шкаф 1".
14. Обновить год издания всех книг, написанных автором с именем "Анна", на 2023 год.
15. Обновить название раздела всех книг, изданных в городе "Санкт-Петербург", на "Классика".
16. Обновить год издания всех книг, написанных автором с фамилией "Петров", на 2024 год.

SQL-запросы на удаление данных из БД:

1. Удалить все записи из таблицы Книги, у которых Раздел = 'Фантастика':
2. Удалить все записи из таблицы Книги, у которых ГодИздания меньше 2000:
3. Удалить все записи из таблицы АвторКниги, у которых КодАвтора равен 1:
4. Удалить все записи из таблицы Авторы, у которых Фамилия начинается с буквы "А":
5. Удалить все записи из таблицы Издательства, у которых Город равен "Москва":
6. Удалить все записи из таблицы АвторКниги, у которых КодКниги равен 10:
7. Удалить все записи из таблицы Книги, у которых МестоХранения равно "Склад":
8. Удалить все записи из таблицы Разделы, у которых Раздел равен "Детективы":
9. Удалить все записи из таблицы АвторКниги, у которых КодАвтора равен 2:
10. Удалить все записи из таблицы Издательства, у которых Издательство равно "O'Reilly Media":
11. Удалить все записи из таблицы Книги, у которых Название содержит слово "Война":
12. Удалить все книги, которые были изданы до 2000 года включительно, и которые хранятся в месте хранения "Библиотека №1".
13. Удалить всех авторов, у которых нет книг в таблице Книги.
14. Удалить все книги, изданные в городе "Москва", из таблицы "Книги".
15. Удалить всех авторов, чьи фамилии начинаются на букву "А" из таблицы "Авторы" и соответствующие записи из таблицы "АвторКниги".
16. Удалить все записи из таблицы "АвторКниги", связанные с книгами, изданными в городе "Москва".

17. Удалить все книги из таблицы "Книги", которые были написаны авторами, чьи фамилии начинаются на букву "П".
18. Удалить все книги из таблицы "Книги", которые были изданы в городах, название которых начинается на букву "Н".
19. Удалить все записи из таблицы "АвторКниги", связанные с книгами, изданными в городах, название которых начинается на букву "Н".

Вариант 6

Разработать БД Аптека содержащую следующие таблицы. Установить связь между таблицами. Заполнить таблицы произвольными данными (не менее 10 записей). Реализовать SQL-запросы на выборку, обновление, удаление данных из БД.

Структура таблицы "Лекарственные средства":

Код препарата (уникальный идентификатор препарата)

Название препарата

Применение

Страна-Производитель

Цена

Структура таблицы "Наличие на складе":

Код препарата (ссылка на таблицу "Лекарственные средства")

Количество

Дата использования

Структура таблицы "Аптечный пункт":

Номер пункта (уникальный идентификатор аптечного пункта)

Адрес

Название препарата (ссылка на таблицу "Лекарственные средства")

Заявки

Дата Заявки

Сумма заказа

SQL-запросы на выборку данных из БД:

1. Вывести список всех препаратов с указанием количества их наличия на складе.
2. Вывести список всех препаратов, имеющихся на складе в количестве менее 10 штук.
3. Вывести список всех препаратов, которые производятся в России.
4. Вывести список всех аптечных пунктов с указанием адреса и количества наличия препаратов в каждом пункте.
5. Вывести список всех лекарственных препаратов, цена которых меньше 1000 руб., отсортированных по названию.
6. Вывести список лекарственных препаратов и их наличие на складе в каждом аптечном пункте.
7. Вывести список лекарственных препаратов, которые заканчиваются на складе в определенном аптечном пункте.
8. Вывести список аптечных пунктов, в которых есть хотя бы одно лекарственное средство срок годности которого истекает в этом месяце.

SQL-запросы на выборку данных из БД:

1. Обновить количество препарата "Аспирин" на складе до 100 штук:
2. Изменить дату использования для всех препаратов производства Индии:
3. Увеличить цену препарата "Метформин" производства Германии на 20%:

4. Обновление цены для всех препаратов в таблице "Лекарственные средства" с учетом налога в размере 20%:
2. Обновление наличия препарата на складе в таблице "Наличие на складе" с учетом отгрузки некоторого количества препарата:
3. Обновление наличия препарата на складе в таблице "Наличие на складе" с учетом поставки новых партий препарата:
4. Обновление цены препарата на складе в таблице "Наличие на складе" при изменении цены в таблице "Лекарственные средства":
5. Обновление даты использования препарата в таблице "Наличие на складе" при продаже препарата в аптечном пункте:
6. Обновление данных о препарате в таблице "Лекарственные средства" при добавлении новой партии препарата путем изменения даты производства и срока годности:
7. Обновление данных о препарате в таблице "Лекарственные средства" при изменении наименования препарата:
8. Обновление цены препарата в таблице "Лекарственные средства" при изменении налогового законодательства:
9. Обновление данных о препарате в таблице "Лекарственные средства" при изменении страны-производителя:
10. Обновление данных о препарате в таблице "Лекарственные средства" при изменении срока годности:

SQL-запросы на удаление данных из БД:

1. Удалить все препараты, которые просрочены на дату 2021-09-30:
2. Удалить все препараты производства США с ценой ниже 10 долларов:
3. Удалить все записи о заявках, где были заказаны препараты с ценой выше 50 долларов:
4. Удалить записи о заявках для препаратов, которых нет на складе:
5. Удалить все записи о препаратах производства России:
6. Удалить все записи о заявках, где было заказано более 10 единиц одного препарата:
7. Удалить все записи о препаратах, которые не используются после даты 2021-10-01:
8. Удалить записи о препаратах, цена на которые была установлена выше 100 долларов:
9. Удалить все записи о заявках, где были заказаны препараты, произведенные в Германии:
10. Удалить все записи о препаратах, название которых начинается на букву "А":
11. Удалить записи о заявках для препаратов, которых нет в таблице "Лекарственные средства":
12. Удалить все записи о препаратах, производимых в США и стоимостью выше 50 долларов:
13. Удалить все записи о заявках в аптечных пунктах на препараты, произведенные в России:
14. Удалить все записи о препаратах, цена на которые не указана:
15. Удалить записи о заявках для препаратов, которые отсутствуют на складе: