Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий

Кафедра ИС

# ОТЧЕТ

по лабораторной работе №3

СОЗДАНИЕ СХЕМЫ БД. ССЫЛОЧНАЯ ЦЕЛОСТНОСТЬ

Выполнил:

ст. гр. ИС/б-21-2-о

Мовенко К. М.

Проверил:

Гончаренко Д. Г.

Севастополь

2023

# ЦЕЛЬ РАБОТЫ

Научиться анализировать предметную область с целью создания схемы БД, учитывая ссылочную целостность набора.

# ВАРИАНТ ЗАДАНИЯ

Создать недостающие таблицы для схемы БД, заданной вариантом 12 (Рисунок 2.1). Выделить и классифицировать все существующие связи, определить необходимые ограничения целостности.

Изменить уже созданные таблицы таким образом, чтобы они могли участвовать в связях. Установить связи между таблицами и проверить работу ограничений целостности в полученной БД.



Рисунок 2.1 – Схема базы данных

# ХОД РАБОТЫ

Сначала была проанализирована таблица Person (Персона). Первичным ключом в ней был установлен атрибут p\_id, а на столбец full\_name было наложено ограничение уникальности.

alter table person add primary key (p\_id);

alter table person add unique (full\_name);

alter table person add check (p\_id > 0);

Для таблицы cases (Дело) был установлен первичный ключ на атрибуте c\_num, а для Alias (Псевдоним) – a\_num.

alter table cases add primary key (c\_num);

alter table alias add primary key (a\_num);

alter table cases add check (c\_num > 0);

alter table alias add check (a\_num > 0);

alter table alias add unique(alias);

alter table cases add unique(descr);

Была создана таблица Relatives (Родственник), связывающая людей, указанных в таблице Person, различными родственными связями.

create table relatives(

eld\_id int not null,

jun\_id int not null,

rel\_type varchar(20) not null,

primary key (eld\_id, jun\_id),

foreign key (eld\_id) references person(p\_id) on delete cascade,

foreign key (jun\_id) references person(p\_id) on delete cascade

);

Данная таблица была заполнена рядом значений (Рисунок 3.1).

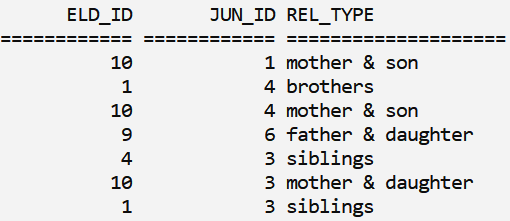


Рисунок 3.1 – Заполнение таблицы Relatives

Была создана таблица Person\_Case (Персона\_Дело), связывающая людей из таблицы Person с уголовными делами из таблицы Cases.

create table person\_case(

p\_id int not null,

c\_id int not null,

clause varchar(30) not null,

term int check (term > 0),

conv\_date date not null,

conv\_place varchar(30),

primary key (p\_id, c\_id),

foreign key (p\_id) references person(p\_id) on delete cascade,

foreign key (c\_id) references cases(c\_num) on delete cascade

);

Данная таблица была заполнена рядом значений (Рисунок 3.2).

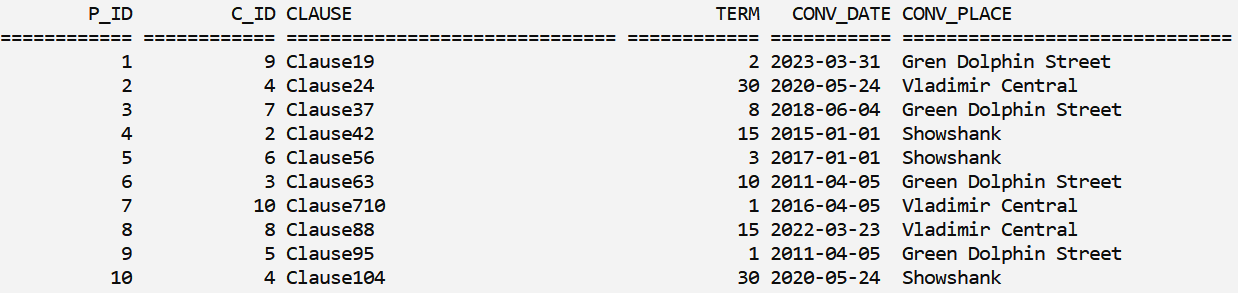


Рисунок 3.2 – Заполнение таблицы Person\_Case

Была создана таблица Person\_Alias (Персона\_Псевдоним), связывающая людей из таблицы Person с псевдонимами из таблицы Alias.

create table person\_alias(

p\_id int not null,

a\_id int not null,

foreign key (p\_id) references person(p\_id) on delete cascade,

foreign key (a\_id) references alias(a\_num) on delete cascade,

primary key (p\_id, a\_id)

);

Данная таблица была заполнена рядом значений (Рисунок 3.3).

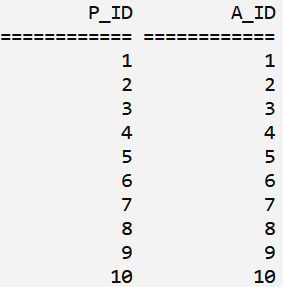


Рисунок 3.3 – Заполнение таблицы Person\_Alias

При попытке внести в таблицу Relatives кортеж с несуществующим в таблице Person значением ключа p\_id система выдала ошибку (Рисунок 3.4).

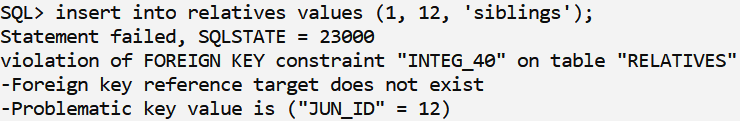


Рисунок 3.4 – Проверка ограничения внешнего ключа

После удаления из таблицы Alias кортежа со значением a\_num = 10 из таблицы Person\_Alias был удалён кортеж, ссылающийся на него (Рисунок 3.5).

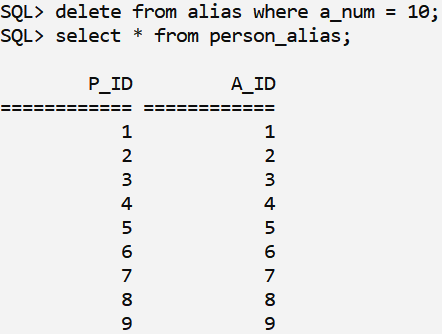


Рисунок 3.5 – Проверка каскадного удаления по ключу

При попытку ввести в таблицу Relatives кортеж, первичные ключи которого совпадают с уже существующим в ней кортежем, система вывела сообщение об ошибке (Рисунок 3.6).

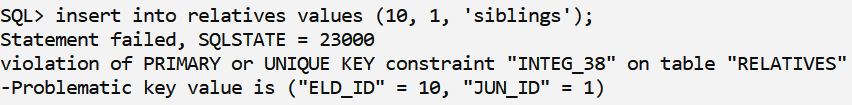


Рисунок 3.6 – Проверка ограничения первичного ключа

# КОНТРОЛЬНЫЕ ВОПРОСЫ

## Требования к ссылочной целостности данных

Требование ссылочной целостности для БД состоит в том, что для каждого значения внешнего ключа, появляющегося в дочернем отношении, в родительском должен найтись кортеж с таким же значением первичного ключа.

Также при удалении кортежа с первичным ключом нельзя, чтобы в дочернем отношении оставались ключи, ссылающиеся на него.

## Типы связей между отношениями

1:1 – ключевой столбец одной таблицы присутствует в другой таблицы, где для него задана уникальность. Наиболее редкий тип связи.

1:М – ключевой таблиц одной таблицы присутствует в другой, где может упоминаться множество раз.

М:М – нескольким записям из одной таблицы может соответствовать множество записей другой таблицы. В SQL реализуется через отдельную таблицу с двумя полями 1:М.

## Стандартная нотация ER-диаграмм

Сущность - прямоугольник, над ним имя сущности. Одна сущность – одна таблица. Разделяется линией на две части: ключевые и обычные атрибуты.

Связь между двумя отношениями изображается с помощью линии. Идентифицирующая связь изображается сплошной линией, не идентифицирующая - пунктирной. Арность связи указывается так: со стороны «многие» ставится жирная точка, со стороны «один» точка не ставится.

Категории – сущность, обобщающая данные для двух похожих таблиц.

## Нормальные формы для баз данных

Нормальная форма — требование, предъявляемое к структуре таблиц в теории реляционных баз данных для устранения из базы избыточных функциональных зависимостей между атрибутами (полями таблиц).

Существует восемь нормальных норм БД, отличных своими ограничениями.

## Необходимость процесса нормализации БД

Цель нормализации БД — исключить избыточное дублирование данных, которое является причиной аномалий, возникающих при добавлении, редактировании и удалении кортежей (строк таблицы).

## Приведение БД к нормальной форме Бойса-Кодда

Нормальная форма Бойса-Кодда – промежуточная между 3 и 4 или усиленная третья нормальная форма.

Для удовлетворения ей таблица должна находится в третьей нормальной форме, а ключевые атрибуты составного ключа не зависели от неключевых атрибутов.

## В какой нормальной форме находится полученная схема БД?

Полученная схема БД находится в первой нормальной форме, так как все атрибуты в ней являются простыми, не ключевые атрибуты не зависят от первичных ключей. Исключено повторение строк в таблицах.

## Способы повышения надежности данных

Методы повышения надёжности данных: репликация данных и резервное копирование, использование транзакций и кластеризации, обновление ПО и мониторинг базы данных, система контроля версий и защита от несанкционированного доступа, конфигурация системы базы данных с учетом производительности и отказоустойчивости, обучение пользователей БД.

# ВЫВОД

В ходе работы были изучены типы связей между отношениями в БД, а также методы поддержания целостности информации в базах данных. В результате был создан ряд таблиц, содержащих внешние ключи и различные виды ограничений.