

ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

старший преподаватель

должность, уч. степень, звание

подпись, дата

Н. В. Путилова

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

Разработка физической модели базы данных с учетом декларативной
ссылочной целостности

по курсу: Проектирование баз данных

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4936

15.09.21

подпись, дата

М.Р. Назаров

инициалы, фамилия

Санкт-Петербург 2021

Цель работы

Создать физическую модель базы данных, находящуюся в третьей нормальной форме в соответствии с заданным вариантом. Расписать ссылочную целостность БД в таблице.

Теоретическая часть

Физические модели данных (PDM) обычно генерируются из логических моделей, но могут быть и воссозданы из действующей базы данных. Средства физического моделирования поддерживают множество стандартных нотаций, а также документирование, генерацию кода и воссоздание структур данных для более чем 70 РСУБД (в том числе новейших версий СУБД Oracle, IBM, Microsoft, Sybase, Teradata, MySQL, HP NeoView и многих других). Поддерживаются все объекты СУБД, необходимые для связывания таблиц и достижения требуемой производительности, такие как индексы, констрейнты, партиции и кластеры, а также новые технологии, такие как Java, XML и веб-сервисы в БД; проектируются средства обеспечения безопасности, применяются передовые методы создания представлений, обеспечивается ряд других возможностей. Физическую модель данных, как правило, можно использовать для оценки объема будущей БД; она может содержать специфическую информацию о выделении дисковой памяти для заданной СУБД.

1. Можно выделить следующие этапы методологии физического проектирования баз данных.
2. Перенос глобальной логической модели данных в среду целевой СУБД.
3. Проектирование основных отношений.
4. Разработка способов получения производных данных.
5. Реализация ограничений предметной области.
6. Проектирование физического представления базы данных.
7. Анализ транзакций.
8. Выбор файловой структуры.
9. Определение индексов.
10. Определение требований к дисковой памяти.
11. Проектирование пользовательских представлений.
12. Разработка механизмов защиты.
13. Обоснование необходимости введения контролируемой избыточности.
14. Текущий контроль и настройка операционной систем
15. Логическая модель расширяет концептуальную путем определения для сущностей их атрибутов, описаний и ограничений, уточняет состав сущностей и взаимосвязи между ними.
16. Физическая модель данных описывает реализацию объектов логической модели на уровне объектов конкретной базы данных.

Логическое представление реляционных баз данных упрощается созданием связей между данными на основе (логической) конструкции, называемой таблицей. Под таблицей понимается двумерная структура, состоящая из строк и столбцов. Пользователь должен понимать, что таблица содержит группу связанных сущностей, т. е. набор сущностей; по этой причине термины набор сущностей и таблица чаще всего означают одно и то же. Таблица также называется отношением (relation), поскольку создатель реляционной модели Э. Ф. Кодд (E. F. Codd) использовал термин "отношение" как синоним слова "таблица".

Правила целостности.

ЦЕЛОСТНОСТЬ НА УРОВНЕ СУЩНОСТИ

Требование: Все элементы первичного ключа уникальны и никакая часть первичного ключа не может быть пустой (null).

Назначение: Гарантирует, что каждая сущность (логический объект) будет иметь уникальную идентификацию, а значения внешнего ключа могут должным образом ссылаться на значения первичного ключа.

Пример: Счет не может иметь несколько дублирующихся значений и не может иметь пустое значение (null). Короче говоря, все счета уникально идентифицируются своим номером.

ЦЕЛОСТНОСТЬ НА УРОВНЕ ССЫЛКИ

Требования: Внешний ключ может иметь или пустое значение (если только он не является частью первичного ключа данной таблицы), или значение, совпадающее со значением первичного ключа в 8 связанной таблице. (Каждое непустое значение внешнего ключа должно ссылаться на существующее значение первичного ключа.)

Назначение: Допускается, что атрибут не имеет соответствующего значения, но атрибут не может принимать недопустимые значения. Выполнение правила целостности на уровне ссылки делает невозможным удаление строки в одной таблице, где первичный ключ имеет обязательное соответствие со значением внешнего ключа в другой таблице.

Пример: Клиенту может быть не назначен (еще) торговый агент, но невозможно назначить клиенту несуществующего агента.

Теперь о внешних ключах:

— Если сущность С связывает сущности А и В, то она должна включать внешние ключи, соответствующие первичным ключам сущностей А и В.

— Если сущность В обозначает сущность А, то она должна включать внешний ключ, соответствующий первичному ключу сущности А.

— Здесь для обозначения любой из ассоциируемых сущностей (стержней, характеристик, обозначений или даже ассоциаций) используется новый обобщающий термин "Цель" или "Целевая сущность".

— Таким образом, при рассмотрении проблемы выбора способа представления ассоциаций и обозначений в базе данных основной вопрос, на который следует получить ответ: "Каковы внешние ключи?". И далее, для каждого внешнего ключа необходимо решить три вопроса:

1. Может ли данный внешний ключ принимать неопределенные значения (NULL-значения)? Иначе говоря, может ли существовать некоторый экземпляр сущности данного типа, для которого неизвестна целевая сущность, указываемая внешним ключом? В случае поставок это, вероятно, невозможно – поставка, осуществляемая неизвестным поставщиком, или поставка неизвестного продукта не имеют смысла. Но в случае с сотрудниками такая ситуация однако могла бы иметь смысл – вполне возможно, что какой-либо сотрудник в данный момент не зачислен вообще ни в какой отдел. Заметим, что ответ на данный вопрос не зависит от прихоти проектировщика базы данных, а определяется фактическим образом действий, принятым в той части реального мира, которая должна быть представлена в рассматриваемой базе данных. Подобные замечания имеют отношение и к вопросам, обсуждаемым ниже.

2. Что должно случиться при попытке УДАЛЕНИЯ целевой сущности, на которую ссылается внешний ключ? Например, при удалении поставщика, который осуществил по крайней мере одну поставку. Существует три возможности: КАСКАДИРУЕТСЯ Операция удаления "каскадируется" с тем, чтобы удалить также поставки этого поставщика.

ОГРАНИЧИВАЕТСЯ Удаляются лишь те поставщики, которые еще не осуществляли поставок. Иначе операция удаления отвергается. УСТАНОВЛИВАЕТСЯ Для всех поставок удаляемого поставщика NULL-значение внешний ключ устанавливается в неопределенное значение, а затем этот поставщик удаляется. Такая возможность, конечно, неприменима, если данный внешний ключ не должен содержать NULL-значений.

3. Что должно происходить при попытке ОБНОВЛЕНИЯ первичного ключа целевой сущности, на которую ссылается некоторый внешний ключ? Например, может быть предпринята попытка обновить номер такого поставщика, для которого имеется по крайней мере одна соответствующая поставка. Этот случай для определенности снова рассмотрим подробнее. Имеются те же три возможности, как и при удалении:

КАСКАДИРУЕТСЯ Операция обновления "каскадируется" с тем, чтобы обновить также и внешний ключ в поставках этого поставщика.

ОГРАНИЧИВАЕТСЯ Обновляются первичные ключи лишь тех поставщиков, которые еще не осуществляли поставок. Иначе операция обновления отвергается.

УСТАНОВЛИВАЕТСЯ Для всех поставок такого поставщика NULL-значение внешний ключ устанавливается в неопределенное значение, а затем обновляется первичный ключ поставщика. Такая возможность, конечно, неприменима, если данный внешний ключ не должен содержать NULL-значений.

— Таким образом, для каждого внешнего ключа в проекте проектировщик базы данных должен специфицировать не только поле или комбинацию полей, составляющих этот внешний ключ, и целевую таблицу, которая идентифицируется этим ключом, но также и ответы на указанные выше вопроса (три ограничения, которые относятся к этому внешнему ключу).

Описание может быть не в таблице, но должно содержать следующие данные:

- Дочерняя таблица;
- Столбцы, составляющие внешний ключ;
- Родительская таблица;
- Наименование ссылочной целостности при удалении;
- Описание действий по поддержанию ссылочной целостности при удалении;
- Наименование ссылочной целостности при обновлении;
- Описание действий по поддержанию ссылочной целостности при обновлении;
- Обоснование выбора типа поддержки ссылочной целостности.

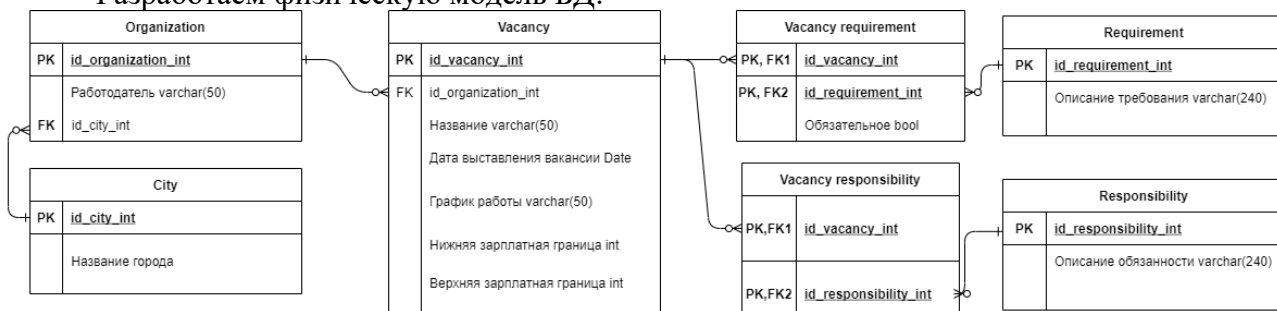
Вариант лабораторной работы

Вариант 13. Вакансии: волонтерские позиции, название вакансии, организация работодатель, адрес работодателя, диапазон зарплаты, требования к образованию, Обязанности, график работы, требования обязательные, желательные, дата выставления вакансии.

- вакансии, имеющие в названии SQL;
- работодатели в Санкт-Петербурге, выставившие вакансии программиста и системного администратора;
- вакансия с наименьшей зарплатой;
- вакансии с минимальным количеством обязательных требований;
- вакансии, в которых нет требования к опыту работы.

Ход работы

Разработаем физическую модель БД:



Опишем ссылочную целостность:

Дочерняя таблица (с внешним ключом)	Внешний ключ	Родительская таблица	Как поддерживается ссылочная целостность при	Описание ссылочной целостности при удалении	Как поддерживается ссылочная целостность при	Описание ссылочной целостности при обновлении	Обоснование

			удалении		обновлении		
Organization	id_city_int	City	Устанавливается	При удалении данных из таблицы «City», значения ссылающихся на них данных из таблицы «Organization» получают «null»	Каскадируется	При обновлении и данных из таблицы «City», обновятся данные в таблице «Organization»	При удалении города компании должны будут перемещены в другой город, организация продолжит существовать. Если id города будет изменен, то он будет изменен и в организации.
Vacancy	id_organization_int	Organization	Каскадируется	При удалении данных из таблицы «Organization» удаляются данные в таблице «Vacancy»	Каскадируется	При обновлении и данных из таблицы «Organization», обновятся данные в таблице «Vacancy»	При удалении организации удалятся и вакансии в организации. При обновлении id организации должен обновиться и id в вакансии
Vacancy requirement	id_vacancy_int	Vacancy	Устанавливается	При удалении данных из таблицы «Vacancy», значения ссылающихся	Каскадируется	При обновлении и данных из таблицы «Vacancy», обновятся	При удалении вакансии, требования остаются, но связь с

				хся на них данных из таблицы «Vacancy requirement» получают «null»		данные в таблице «Vacancy requirement»	удаленной вакансией обнуляется. При обновлении вакансии обновляется и связь с требованием
Vacancy requirement	id_requirement_int	Requirement	Устанавливается	При удалении данных из таблицы «Requirement», значения ссылающихся на них данных из таблицы «Vacancy requirement» получают «null»	Каскадируется	При обновлении данных из таблицы «Requirement», обновятся данные в таблице «Vacancy requirement»	При удалении требования, вакансии остаются, но связь с удаленным требованием обнуляется. При обновлении требования обновляется и связь с вакансией
Vacancy responsibility	id_vacancy_int	Vacancy	Устанавливается	При удалении данных из таблицы «Vacancy», значения ссылающихся на них данных из таблицы «Vacancy responsibility» получают «null»	Каскадируется	При обновлении данных из таблицы «Vacancy», обновятся данные в таблице «Vacancy responsibility»	При удалении вакансии, обязанности остаются, но связь с удаленной вакансией обнуляется. При обновлении вакансии обновляет

							ся и связь с требованием
Vacancy responsibility	id_responsibility_int	Responsibility	Устанавливается	При удалении данных из таблицы «Responsibility», значения ссылающиеся на них данных из таблицы «Vacancy responsibility» получают «null»	Каскадируется	При обновлении данных из таблицы «Responsibility», обновятся данные в таблице «Vacancy responsibility»	При удалении обязанности, вакансии остаются, но связь с удаленной обязанностью обнуляется. При обновлении обязанности обновляется и связь с вакансией

Вывод

В ходе лабораторной работы были созданы физическая модель базы данных, проведена ее нормализация и была расписана ссылочная целостность БД.