Содержание:

Введение

Рассмотрим пример учебной базы данных. Имеется отдел, который занимается обработкой звонков, поступающих на различные линии. Линии обслуживаются конкретными операторами. Операторы состоят в разных группах под присмотром супервайзеров.

Только из данного краткого описания можно выделить несколько самостоятельных объектов:

- Телефонные линии обслуживания;
- Сотрудники отдела;
- Должности сотрудников;
- Группы, по которым распределены сотрудники;
- Звонки.

Ознакомившись с диаграммой базы данных, можно обратить внимание на то, что некоторая информация из одних таблиц присутствует в других, т.е. между ними имеются связи.

В нашем конкретном случае, все таблицы можно соединить между собой. Чтобы понять, как это правильно сделать, необходимо рассмотреть типы связей.

Логику соединения таблиц в БД важно понять с самого начала изучения SQL, так как наверняка Вы не будете писать запросы только к одной таблице.

Всего существует 3 типа связей:

- Один к одному;
- Один ко многим;
- Многие ко многим.

Связь «Один к одному»

Связь один к одному образуется, когда ключевой столбец (идентификатор) присутствует в другой таблице, в которой тоже является ключом либо свойствами столбца задана его уникальность (одно и тоже значение не может повторяться в разных строках).

На практике связь «один к одному» наблюдается не часто. Например, она может возникнуть, когда требуется разделить данных одной таблицы на несколько отдельных таблиц с целью безопасности.

В учебной безе данных нет подходящего примера, но гипотетически могла бы существовать необходимость разделения таблицы сотрудников.

Пример:
Представьте, что базой данных пользуются несколько менеджеров и аналитиков, а таблица «Сотрудники» содержит те же столбцы, что и учебная база. Следовательно, доступ к персональным данным может получить любой из упомянутых работников.

Чтобы устранить возможность утечки конфиденциальной информации, принимается решение о переносе информации паспортных данных в отдельную таблицу, доступ к которой предоставляется ограниченному кругу лиц.

Связь «Один ко многим»

В типе связей один ко многим одной записи первой таблицы соответствует несколько записей в другой таблице.

Рассмотрим связь учебной базы данных между должностями и сотрудниками, которая относится к рассматриваемому типу.

Записи должностей в таблице «Должность» уникальны, так как нет смысла повторно создавать имеющуюся запись. Записи в таблице «Сотрудники» также уникальны, но несколько различных сотрудников могут находиться на одинаковой должностной позиции.

Символ ключа на конце связи указывает, что таблица, к которой этой конец прилегает, находится на стороне «один» (связанный столбец является первичным

ключом), а символ бесконечности находится на стороне «многие» (такой столбец является внешним ключом).

Связь «Многие ко многим»

Если нескольким записям из одной таблицы соответствует несколько записей из другой таблицы, то такая связь называется «многие ко многим» и организовывается посредством связывающей таблицы.

В нашей базе подобное наблюдается только между таблицами с сотрудниками и линиями.

В большинстве случаев любые две таблицы связаны отношением «один-ко-многим». Это означает, что любая запись в первой таблице может быть связана с несколькими записями во второй, однако любая запись второй таблицы связана только с одной записью в первой.

Иногда возникает потребность разбить одну таблицу на более мелкие – проблема может заключаться в том, что некоторые сведения из нее используются не слишком часто, или в том, что какие-то данные не предназначаются для всеобщего доступа. Например, часть информации о факультетах нужна только для рекламных целей и используется очень редко. С другой стороны, сведения о заработной плате должны быть доступны только определенным сотрудникам. В любом из этих случаев можно создать отдельную таблицу и связать ее с исходной таблицей отношением типа «один-к-одному». Это означает, что любая запись в первой таблице связана только с одной записью во второй.

Запрос — объект базы данных, используемый для выборки или модификации хранимых данных.

Запрос на выборку является наиболее часто используемым типом запроса. Запросы этого типа выбирают данные из одной или нескольких таблиц и отображают их в виде таблицы, записи в которой можно обновлять (с некоторыми ограничениями). Запросы на выборку можно также использовать для группировки записей и вычисления сумм, средних значений, подсчета записей и нахождения других типов итоговых значений.

Для подготовки запросов используются:

- QBE (Query By Example) — язык запросов по образцам,
- SQL (Structured Query Language) — язык структурированных запросов.

Sql- язык структурированных запросов

Язык структурированных запросов SQL является наиболее распространенным языком управления базами данных клиент/сервер. содержит операторы:

- описания данных (DDL — Data Definition Language). Основные операторы DDL — Create Domain (Создание домена), Alter Domain (Изменение домена), Drop Domain (Уничтожение домена), Create Table, Alter Table, Table Drop;
- управления данными (DML — Data Manipulation Language). Основные операторы DML — Select (Выбор), Insert (Вставка), Update (Обновление), Delete (Удаление);
- формирования запросов.

Операторы DDL (Data Definition Language) - операторы определения объектов базы данных

- - CREATE SCHEMA - создать схему базы данных
  - DROP SHEMA - удалить схему базы данных
  - CREATE TABLE - создать таблицу
  - ALTER TABLE - изменить таблицу
  - DROP TABLE - удалить таблицу
  - CREATE DOMAIN - создать домен
  - ALTER DOMAIN - изменить домен
  - DROP DOMAIN - удалить домен
  - CREATE COLLATION - создать последовательность
  - DROP COLLATION - удалить последовательность
  - CREATE VIEW - создать представление
  - DROP VIEW - удалить представление

Операторы DML (Data Manipulation Language) - операторы манипулирования данными

- - - SELECT - отобрать строки из таблиц
    - INSERT - добавить строки в таблицу
    - UPDATE - изменить строки в таблице
    - DELETE - удалить строки в таблице
    - COMMIT - зафиксировать внесенные изменения
    - ROLLBACK - откатить внесенные изменения

Ключевые слова sql

1. **Команды** – представляют собой глаголы определяющего действия, которые необходимо выполнить (SELECT, ALTER, CREATE, DROP)
2. **Условия** (квалификатор)-ограничивают диапазон значений элементов, входящих в запрос (WHERE)
3. **Модификаторы** (предложения)-модифицируют выполнение инструкций (ORDER BY)
4. **Предикаты** – представляют собой выражения, такие как IN. Могут возвращать в качестве результата значения True, False, в некоторых случаях Null (неизвестный результат)
5. **Операторы** (=, <,>)- сравнивают значения или применяются для создания объединений в синтаксисе предложений WHERE. Эти операторы наз предикатами сравнения.
6. **Статистические функции** (агрегаты)- возвращают одно результирующее значение на основе набора данных (SUM (сумма), COUNT (количество), MIN (минимальное значение), MAX (максимальное значение) или AVG (среднее значение)
7. **Функции преобразования типа данных** – изменяют тип данных с одного на др.(CAST, CONVERT)
8. **Другие ключевые слова** (зарезервированные)-изменяющие действие команд или управляющие курсором (указателем текущей записи в наборе)

В синтаксических конструкциях используются следующие обозначения:

звездочка (*) для обозначения "все" - употребляется в обычном для программирования смысле, т.е. "все случаи, удовлетворяющие определению";

квадратные скобки ([]) – означают, что конструкции, заключенные в эти скобки, являются необязательными (т.е. могут быть опущены);

фигурные скобки ({}) – означают, что конструкции, заключенные в эти скобки, должны рассматриваться как целые синтаксические единицы, т.е. они позволяют уточнить порядок разбора синтаксических конструкций, заменяя обычные скобки, используемые в синтаксисе SQL;

многоточие (...) – указывает на то, что непосредственно предшествующая ему синтаксическая единица факультативно может повторяться один или более раз;

прямая черта (|) – означает наличие выбора из двух или более возможностей. Например обозначение ASC|DESC указывает, можно выбрать один из терминов ASC или DESC; когда же один из элементов выбора заключен в квадратные скобки, то это означает, что он выбирается по умолчанию (так, [ASC]|DESC означает, что отсутствие всей этой конструкции будет восприниматься как выбор ASC);

точка с запятой (;) – завершающий элемент предложений SQL;

запятая (,) – используется для разделения элементов списков;

пробелы ( ) – могут вводиться для повышения наглядности между любыми синтаксическими конструкциями предложений SQL;

прописные жирные латинские буквы и символы – используются для написания конструкций языка SQL и должны (если это специально не оговорено) записываться в точности так, как показано;

строчные буквы – используются для написания конструкций, которые должны заменяться конкретными значениями, выбранными пользователем, причем для определенности отдельные слова этих конструкций связываются между собой символом подчеркивания (_);

Основой SQL является инструкция SELECT, используемая для создания запросов на выборку.

Синтаксис инструкции:

SELECT [ALL │ DISTINCT │DISTINCTROW] список_выбора

[AS псевдоним 1[, псевд 2 [,…]]]

FROM имена таблиц

[WHERE критерий поиска]

[GROUP BY имя столбца, имя столбца,…]

[HAVING условие поиска]

[ORDER BY критерий столбца [ASC│DESC]];

SELECT — выбрать (директива) данные из указанных столбцов и (если необходимо) выполнить перед выводом их преобразование в соответствии с указанными

выражениями и (или) функциями

FROM — из (условие) перечисленных таблиц, в которых расположены эти столбцы

WHERE — где (условие) строки из указанных таблиц должны удовлетворять указанному перечню условий отбора строк

GROUP BY — группируя по (условие) указанному перечню столбцов с тем, чтобы получить для каждой группы единственное агрегированное значение, используя во фразе SELECT SQL – функции: SUM (сумма), COUNT (количество), MIN (минимум), MAX (максимум), AVG (среднее значение)

HAVING — имея в результате лишь те группы, которые удовлетворяют указанному перечню условий отбора групп (условие)

ORDER BY — спецификация сортировки (условие) определяет порядок сортировки: ASC – сортировка по возрастанию, DESC - сортировка по убыванию.

ПРЕДИКАТЫ :

1. Сравнения =, <>, >=, <, <=

2. В интервале - “между” BETWEEN a1 and a2

3. Входит в множество IN (=[Товар] IN (“Мука”,“Крупа”.....)

4. Подобие < имя > Like < образец > (что) (с чем сравнивать)

Для чего все это нужно?

Связи выполняют более важную роль, чем просто информация размещения данных по таблицам. Прежде всего они требуются разработчикам для поддержания целостности баз данных.

Правильно настроив связи, можно быть уверенным, что ничего не потеряется.

Источники:

https://skarlupka.ru/articles.php?id=26

https://www.yaklass.ru/materiali?mode=cht&chtid=511

https://zametkinapolyah.ru/zametki-o-mysql/chast-3-2-vidy-svyazej-mezhdu-tablicami-v-baze-dannyx-svyazi-v-relyacionnyx-bazax-dannyx-otnosheniya-kortezhi-atributy.