### Инвариантная часть

**Задание 1.1: Заполните таблицу «Типы данных и объекты СУБД MySQL**

|  |  |  |
| --- | --- | --- |
| **№** | **Тип данных / Объекты СУБД MySQL** | **Описание / Характеристики** |
| 1 | BLOB | **Binary Large Object**. Двоичные объекты большого размера. Используются для хранения изображений, аудио, видео и других нетекстовых данных. Существует несколько подтипов (TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB) с разными максимальными размерами. Размер BLOB до 65,535 байт. |
| 2 | DECIMAL | Числа с фиксированной точкой. Используются для хранения финансовых данных или других значений, требующих высокой точности. При объявлении указываются точность (общее количество цифр) и масштаб (количество цифр после десятичной точки). Например, DECIMAL(10,2) может хранить числа от -99999999.99 до 99999999.99. |
| 3 | INT | **Integer**. Целочисленный тип данных. Используется для хранения целых чисел. Диапазон значений зависит от размера: TINYINT (1 байт), SMALLINT (2 байта), MEDIUMINT (3 байта), INT (4 байта), BIGINT (8 байт). INT в основном используется для идентификаторов (например, автоинкрементные ID) и переключателей (0 - нет, 1 - да). |
| 4 | DATE | Дата. Хранится в формате ‘YYYY-MM-DD’. Подходит для хранения дат без времени. Диапазон дат: ’1000-01-01’ to ‘9999-12-31’. |
| 5 | TIME | Время. Хранится в формате ‘HH:MM:SS’. Подходит для хранения времени без даты. Диапазон времени: ‘-838:59:59’ to ’838:59:59’. |
| 6 | INDEX | Индекс. Специальная структура данных, создаваемая для одного или нескольких столбцов таблицы. Индексы значительно ускоряют операции поиска и сортировки данных по индексированным столбцам, но замедляют операции записи (INSERT, UPDATE, DELETE). Важно правильно выбирать столбцы для индексирования. |
| 7 | PRIMARY KEY | Первичный ключ. Уникальный идентификатор записи в таблице. Каждая таблица может иметь только один первичный ключ. Значение первичного ключа не может быть NULL. Обычно используется целочисленный тип с автоинкрементом (AUTO\_INCREMENT) для автоматического присвоения уникальных ID новым записям. |
| 8 | FOREIGN KEY | Внешний ключ. Столбец в одной таблице, который ссылается на PRIMARY KEY в другой таблице. Устанавливает связь между таблицами. Обеспечивает ссылочную целостность данных, предотвращая добавление записей с несуществующими значениями в связанной таблице. |
| 9 | FLOAT | Числа с плавающей точкой одинарной точности (4 байта). Используются для хранения чисел с дробной частью, когда высокая точность не требуется. Подвержены ошибкам округления. |
| 10 | DOUBLE | Числа с плавающей точкой двойной точности (8 байт). Используются для хранения чисел с дробной частью, когда требуется более высокая точность, чем у FLOAT. Все еще подвержены ошибкам округления, но в меньшей степени, чем FLOAT. |
| 11 | TRIGGER | Триггер. Автоматически выполняемая процедура (блок SQL-кода) в ответ на определенное событие (INSERT, UPDATE, DELETE) в таблице. Триггеры используются для обеспечения целостности данных, аудита изменений, выполнения сложных бизнес-правил. |
| 12 | TINYINT | Целочисленный тип данных, занимающий 1 байт. Диапазон значений: от -128 до 127 (если signed) или от 0 до 255 (если unsigned). Часто используется для хранения небольших целых чисел, например, кодов статусов, флагов, небольших счетчиков. |
| 13 | BIGINT | Целочисленный тип данных, занимающий 8 байт. Диапазон значений: от -263 до 263-1 (если signed) или от 0 до 264-1 (если unsigned). Используется для хранения очень больших целых чисел, которые не помещаются в INT. |
| 14 | TEXT | Строковый тип данных переменной длины. Используется для хранения длинных текстовых строк. Максимальный размер: 65,535 байт. Существуют подтипы (TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT) с разными максимальными размерами. |
| 15 | TIMESTAMP | Дата и время. Хранится в формате ‘YYYY-MM-DD HH:MM:SS’. Автоматически обновляется при изменении записи, если это указано в определении столбца (ON UPDATE CURRENT\_TIMESTAMP). Используется для отслеживания времени последнего изменения записи. |
| 16 | BOOLEAN | Логический тип данных. Хранит только два значения: TRUE (1) или FALSE (0). В MySQL BOOLEAN является синонимом TINYINT(1). |
| 17 | ENUM | Перечислимый тип данных. Позволяет выбрать одно значение из предопределенного списка строк. Экономит место для хранения, т.к. строки хранятся в виде индексов. |
| 18 | SET | Набор значений. Позволяет выбрать несколько значений из предопределенного списка строк. Каждое значение в SET хранится как бит, поэтому максимальное количество элементов в SET ограничено 64. |
| 19 | VARBINARY | Двоичный строковый тип данных переменной длины. Аналогичен VARCHAR, но предназначен для хранения двоичных данных, а не текста. Полезен для хранения зашифрованных данных или других двоичных данных, длина которых может варьироваться. |
| 20 | VIEW | Представление (View). Виртуальная таблица, представляющая результат запроса SELECT. Представления упрощают сложные запросы, предоставляют уровень абстракции над базовыми таблицами и позволяют контролировать доступ к данным. Изменения в базовых таблицах отражаются в представлении. |
| 21 | CHAR | Текстовый формат фиксированной длины. Используется для хранения строк фиксированной длины. Если строка короче заданной длины, она дополняется пробелами. Максимальная длина: 255 символов (в старых версиях MySQL было ограничение в 255, но в современных версиях, если включена поддержка innodb\_large\_prefix, можно до 3072). Лучше использовать VARCHAR, если длина строки варьируется. |
| 22 | BINARY | Двоичный строковый тип данных фиксированной длины. Аналогичен CHAR, но предназначен для хранения двоичных данных. Если данные короче заданной длины, они дополняются нулями. |
| 23 | YEAR | Год. Хранится в формате ‘YYYY’ (4 цифры) или ‘YY’ (2 цифры). |
| 24 | TIME | Время (Повтор - уже есть в строке 5). Хранится в формате ‘HH:MM:SS’. Подходит для хранения времени без даты. Диапазон времени: ‘-838:59:59’ to ’838:59:59’. |
| 25 | DATETIME | Дата и время. Хранится в формате ‘YYYY-MM-DD HH:MM:SS’. Подходит для хранения дат и времени вместе. |
| 26 | MEDIUMINT | Целочисленный тип данных, занимающий 3 байта. Диапазон значений: от -8388608 до 8388607 (если signed) или от 0 до 16777215 (если unsigned). |
| 27 | PROCEDURE | Хранимая процедура. Именованный блок SQL-кода, который может принимать параметры и выполнять последовательность операций. Хранимые процедуры компилируются и хранятся на сервере, что повышает производительность и безопасность. |
| 28 | FUNCTION | Хранимая функция. Аналогична хранимой процедуре, но обязательно возвращает одно значение. Используется для выполнения вычислений и преобразований данных. |
| 29 | CURSOR | Курсор. Объект, позволяющий построчно обрабатывать результат выполнения SQL-запроса. Используется в хранимых процедурах и функциях для навигации по набору данных и выполнения операций над каждой строкой. |