

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Основы проектирования баз данных

ОДОБРЕНО  
Протоколом Методического совета  
СОГБПОУ ВПТ

УТВЕРЖДАЮ  
Протоколом Педагогического совета  
СОГБПОУ ВПТ

«        »                    20   г. №

«        »                    20   г. №

Организация-разработчик: СОГБПОУ ВПТ

Разработчики:

Коростелев М.К., преподаватель СОГБПОУ ВПТ

Рассмотрено на заседании ПЦК профессиональных дисциплин ППССЗ 09.02.03, 09.02.01

Протокол № \_\_\_\_ от « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

Председатель ПЦК    Никитина С.Ю.

# 1 ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОП.08 ОСНОВЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ»

## 1.1. Область применения программы

Программа учебной дисциплины «ОП.08 Основы проектирования баз данных» является частью ППССЗ в соответствии с ФГОС по специальности среднего профессионального образования 09.02.07 Информационные системы и программирование укрупненной группы специальностей 09.00.00 - Информатика и вычислительная техника.

## 1.2. Место учебной дисциплины в структуре ППССЗ

Дисциплина «ОП.08 Основы проектирования баз данных» входит в состав общепрофессионального цикла программы среднего профессионального образования - программы подготовки специалистов среднего звена - по специальности 09.02.07 Информационные системы и программирование.

Изучение данного учебного курса является необходимой основой для последующего изучения дисциплин профессиональной подготовки, а также для прохождения учебной и производственной практик, подготовки студентов к государственной итоговой аттестации.

## 1.3. Цели и задачи учебной дисциплины - требования к результатам освоения учебной дисциплины:

**Цель изучения дисциплины** - приобретение студентами теоретических знаний и практических умений в области основ программирования и баз данных.

### **Задачи дисциплины:**

- изучение вопросов теории баз данных;
- приобретение устойчивых навыков проектирования и реализации реляционных баз данных;
- ознакомление с принципами организации информационных хранилищ, методами количественного и качественного анализа информации для обоснования и принятия управленческих решений;
- выработка практических навыков применения баз данных и прикладных программ в профессиональной деятельности.

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся в ходе освоения профессионального модуля должен:

### **уметь:**

- проектировать реляционную базу данных;
- использовать язык запросов для программного извлечения сведений из баз данных;

### **знать:**

- основы теории баз данных;
- модели баз данных;
- особенности реляционной модели и проектирование баз данных;
- изобразительные средства, используемые в ER-моделировании;
- основы реляционной алгебры;
- принципы проектирования баз данных;
- обеспечение непротиворечивости и целостности данных;
- средства проектирования структур баз данных;

- язык запросов SQL.

Процесс изучения дисциплины направлен на формирование и развитие следующих компетенций:

- выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам (ОК 01);
- осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности (ОК 02);
- работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами (ОК 04);
- осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста (ОК 05);
- использовать информационные технологии в профессиональной деятельности (ОК 09);
- пользоваться профессиональной документацией на государственном и иностранном языках (ОК 10);
- осуществлять сбор, обработку и анализ информации для проектирования баз данных (ПК 11.1);
- проектировать базу данных на основе анализа предметной области (ПК 11.2);
- разрабатывать объекты базы данных в соответствии с результатами анализа предметной области (ПК 11.3);
- реализовывать базу данных в конкретной системе управления базами данных (ПК 11.4);
- администрировать базы данных (ПК 11.5);
- защищать информацию в базе данных с использованием технологии защиты информации (ПК 11.6).

#### **1.4. Количество часов на освоение примерной программы учебной дисциплины:**

максимальной учебной нагрузки обучающегося 68 часов, в том числе: обязательной аудиторной учебной нагрузки обучающегося 59 часов; самостоятельная работа обучающегося 9 часов.

## 2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

### 2.1 Объем учебной дисциплины и виды учебной работы

<b>Вид учебной работы</b>	<b>Объем часов</b>
<b>Максимальная учебная нагрузка (всего)</b>	68
<b>Обязательная аудиторная учебная нагрузка (всего)</b>	59
в том числе:	
теоретическое обучение	29
практические занятия	30
<b>Самостоятельная работа обучающегося (всего)</b>	9
Итоговая аттестация в форме <i>зачета</i>	

## 2.2 Тематический план и содержание учебной дисциплины «ОП.08 Основы проектирования баз данных»

<i>Наименование разделов и тем</i>	<i>Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся, курсовая работа (проект)</i>	<i>Объем часов</i>	<i>Уровень освоения</i>
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>
<b>Тема 1. Основные понятия баз данных</b>	<b>Содержание учебного материала</b>	<b>10</b>	<b>1</b>
	1. Основные понятия теории БД 2. Технологии работы с БД		
	<b>Практические занятия</b> - Нормализация реляционной БД, освоение принципов проектирования БД. - Преобразование реляционной БД в сущности и связи. - Проектирование реляционной БД. Нормализация таблиц.	<b>6</b>	<b>2</b>
	<b>Самостоятельная работа</b> Составление схемы видов связей между объектами: «Студенты-зачетки», Владельцы-Транспорт.	<b>2</b>	<b>3</b>
<b>Тема 2. Взаимосвязи в моделях и реляционный подход к построению моделей</b>	<b>Содержание учебного материала</b>	<b>10</b>	<b>1</b>
	1. Логическая и физическая независимость данных. 2. Типы моделей данных. Реляционная модель данных.		
	3. Реляционная алгебра.		
	<b>Практические занятия</b> - Задание ключей. Создание основных объектов БД - Создание проекта БД. Создание БД. Редактирование и модификация таблиц - Редактирование, добавление и удаление записей в таблице. Применение логических условий к записям. Открытие, редактирование и пополнение табличного файла	<b>6</b>	<b>2</b>
	<b>Самостоятельная работа</b> Создание запросов к СУБД в приложении Microsoft Office Access.	<b>2</b>	<b>3</b>
<b>Тема 3 Этапы проектирования баз данных</b>	<b>Содержание учебного материала</b>	<b>13</b>	<b>1</b>
	1. Основные этапы проектирования БД.		
	2. Концептуальное проектирование БД.		
	3. Нормализация БД.		

	<b>Практические занятия</b>		<b>2</b>
	<ul style="list-style-type: none"> <li>- Создание ключевых полей. Задание индексов. Установление и удаление связей между таблицами.</li> <li>- Проведение сортировки и фильтрации данных. Поиск данных по одному и нескольким полям. Поиск данных в таблице.</li> <li>- Работа с переменными. Заполнение таблицы из файла. Создание запросов, форм, отчетов.</li> </ul>	<b>6</b>	
	<b>Самостоятельная работа</b> Проектирование базы данных, создание запросов, установление связей, проверка работы БД.	<b>2</b>	<b>3</b>
<b>Тема 4 Проектирование структур баз данных</b>	Содержание учебного материала	<b>10</b>	<b>1</b>
	1. Средства проектирования структур БД.		
	2. Организация интерфейса с пользователем.		
	<b>Практические занятия</b> <ul style="list-style-type: none"> <li>- Добавление записей в табличный файл из двумерного массива. Работа с командами ввода-вывода. Использование функций для работы с массивами.</li> <li>- Создание меню различных видов. Модификация и управление меню.</li> <li>- Создание рабочих и системных окон. Добавление элементов управления рабочим окном</li> <li>- Создание файла проекта базы данных. Создание интерфейса входной формы. Использование исполняемого файла проекта БД, приемы создания и управления.</li> <li>- Обработка транзакций. Использование функций защиты для БД.</li> </ul>	<b>7</b>	<b>2</b>
<b>Самостоятельная работа</b> Составление таблицы объектов по предметной области «Учет продажи товара».	<b>2</b>	<b>3</b>	
<b>Тема 5. Организация запросов SQL</b>	Содержание учебного материала	<b>16</b>	<b>1</b>
	1. Основные понятия языка SQL. Синтаксис операторов, типы данных.		
	2. Создание, модификация и удаление таблиц. Операторы манипулирования данными.		
	3. Организация запросов на выборку данных при помощи языка SQL.		
	4. Типовые запросы языка SQL.		
5. Сортировка и группировка данных в SQL.			

	<p><b>Практические занятия</b></p> <ul style="list-style-type: none"> <li>- Создание формы. Управление внешним видом формы.</li> <li>- Задание значений и ограничений поля. Проверка введенного в поле значения.</li> </ul> <p>Отображение данных числового типа и типа дата.</p> <ul style="list-style-type: none"> <li>- Создание и модификация таблиц БД. Выборка данных из БД. Модификация содержимого БД.</li> </ul>	5	2
	<p><b>Самостоятельная работа</b></p> <p>Составление сценариев транзакций при работе реплик БД «Успеваемость группы».</p> <p>Конструирование дополнительных вложенных SQL запросов по БД «Успеваемость группы».</p> <p>Разработка SQL запросов с использованием функций даты и времени.</p>	1	3
	<b>Всего:</b>	<b>68</b>	

Для характеристики уровня освоения учебного материала используются следующие обозначения:

1. - ознакомительный (узнавание ранее изученных объектов, свойств);
2. - репродуктивный (выполнение деятельности по образцу, инструкции или под руководством);
3. - продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач).



### **3 УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ**

#### **3.1 Материально-техническое обеспечение**

Для реализации программы учебной дисциплины должна быть предусмотрена аудитория, оснащенная следующим оборудованием:

- автоматизированные рабочие места для обучающихся
- автоматизированное рабочее место преподавателя
- проектор мультимедийный
- меловая ученическая доска
- интерактивная доска

#### **3.2 Информационное обеспечение обучения**

**Перечень учебных изданий, Интернет-ресурсов, дополнительной литературы**

##### **Основные источники**

- 1 Федорова Г.Н. Основы проектирования баз данных. - М. : Академия
- 2 Кумскова, И. А. Базы данных: учебник для СПО /.- М.: КНОРУС
- 3 Хомоненко А.Д. и др. Базы данных: учебник для ВУЗов. - СПб.: Корона
- 4 Фуфаев Э.В. Разработка и эксплуатация удаленных баз данных- М.: Академия
- 5 Илюшечкин В.М. Основы использования и проектирования баз данных: учебник для СПО. - М.: Юрайт

##### **Дополнительные источники**

- 1 Баженова И.Ю. Основы проектирования приложений баз данных. - М.: Национальный Открытый Университет «ИНТУИТ»

##### **Базы данных, информационно-справочные и поисковые системы**

1. Единая коллекция цифровых образовательных ресурсов - [http ^/school- collection. edu.ru](http://school-collection.edu.ru)
2. Каталог образовательных ресурсов основного среднего образования - <http://arhcity.ru/?page=266/1>
3. Национальный открытый университет ИНТУИТ - <https://www.intuit.ru/>

#### 4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Контроль и оценка результатов освоения учебной дисциплины осуществляется преподавателем в процессе проведения практических выполнения обучающимися индивидуальных заданий.

<i>Результаты обучения</i>	<i>Критерии оценки</i>	<i>Формы и методы оценки</i>
<p><i>Перечень умений, осваиваемых в рамках дисциплины:</i> проектировать реляционную базу данных; использовать язык запросов для программного извлечения сведений из баз данных</p>	<p>«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.</p>	<p>- Тестирование на знание терминологии по теме; - Контрольная работа; - Самостоятельная работа; - Наблюдение за выполнением практического задания (деятельностью студента); - Оценка выполнения практического задания (работы); - Подготовка и выступление с докладом, сообщением, презентацией</p>
<p><i>Перечень знаний, осваиваемых в рамках дисциплины:</i> основы теории баз данных; модели данных; особенности реляционной модели и проектирование баз данных; изобразительные средства, используемые в ER- моделировании; основы реляционной алгебры; принципы проектирования баз данных; обеспечение непротиворечивости и целостности данных; средства проектирования структур баз данных; язык запросов SQL</p>	<p>«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками. «Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки. «Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.</p>	<p>- Подготовка и выступление с докладом, сообщением, презентацией</p>

#### 5 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Успешное изучение курса требует посещения лекций, активной работы на практических занятиях, выполнения всех учебных заданий преподавателя, ознакомления основной и дополнительной литературой. Работа с конспектом лекций предполагает просмотр конспекта в тот же день после занятий, пометку материала

конспекта, который вызывает затруднения для понимания. Попробуйте найти ответы на затруднительные вопросы, используя рекомендуемую литературу. Если самостоятельно не удалось разобраться в материале, сформулируйте вопросы и обратитесь за помощью к преподавателю на консультации или ближайшей лекции.

Регулярно отводите время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам. Для выполнения письменных домашних заданий необходимо внимательно прочитать соответствующий раздел учебника и проработать аналогичные задания, рассматриваемые преподавателем на лекционных занятиях.

Основным методом самостоятельной работы является изучение и учебно-методических материалов, научной литературы, в том числе из сети Интернет, и применение изученного на практике.

Алгоритм работы над каждой темой:

- изучите содержание темы вначале по лекционному материалу, а затем по другим источникам;

- прочитайте дополнительную литературу из списка, предложенного преподавателем;

- выпишите в тетрадь основные категории и персоналии по теме, используя лекционный материал или словари, что поможет быстро повторить материал при подготовке к сдаче темы или экзамена;

- составьте краткий план ответа по каждому вопросу, выносимому на обсуждение на семинарском занятии;

- выучите определения терминов, относящихся к теме;

- продумайте примеры и иллюстрации к ответу по изучаемой теме;

- продумывайте высказывания по темам, предложенным к занятию.

Рекомендации по работе с учебной литературой:

- ознакомьтесь с аннотациями к рекомендованной литературе и определите основной метод изложения материала того или иного источника;

- составьте собственные аннотации к другим источникам на карточках, что поможет при подготовке рефератов, текстов речей, при подготовке к экзамену;

- выберите те источники, которые наиболее подходят для изучения конкретной темы.

### *Практические задания*

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

## Методические рекомендации по разработке SQL запросов с использованием функций даты и времени

Чтобы получить текущую дату и время используется функция `NOW ()`.

Пример:

```
SELECT NOW ()
```

Результат: 2015-09-25 14:42:53

Для получения только текущей даты есть функция `CURDATE ()`.

Пример:

```
SELECT CURDATE ()
```

Результат: 2015-09-25

И функция `CURTIME ()`, которая возвращает только текущее время:

Пример:

```
SELECT CURTIME ()
```

Результат: 14:42:53

Функции `CURDATE ()` и `NOW ()` удобно использовать для добавления в базу данных записей, для которых требуется хранить дату добавления. Например, при добавлении статьи на сайт хорошо бы хранить ее дату публикации. Тогда запрос на добавление статьи в базу будет примерно таким:

```
INSERT INTO posts (icUpost, text_post, date_publication) VALUES (1, 'текст статьи', NOW ());
```

### Прибавление и вычитание дат и времени

Функция `ADDDATE (date, INTERVAL value)` прибавляет к дате `date` значение `value` и возвращает полученное значение. В качестве `value` могут выступать следующие значения:

- `SECOND` — секунды
- `MINUTE` — минуты
- `HOURL` — часы
- `DAY` — дни

- WEEK — недели
- MONTH — месяцы
- QUARTER — кварталы
- YEAR — годы

а также их комбинации:

- MINUTE\_SECOND — минуты и секунды
- HOUR SECOND<sub>наси</sub> — минуты и секунды
- HOURMINUTE — часы и минуты
- DAY\_SECOND — дни, часы, минуты и секунды
- DAY\_MINUTE — дни, часы и минуты

```
SELECT ADDDATE ('2015-09-28 10:30:20', INTERVAL 1 DAY)
```

Результат: 2015-09-29 10:30:20

- DAYHOUR — дни и часы
- YEARMONTH — года и месяцы.

Пример:

или

```
SELECT ADDDATE ('2015-09-28 10:30:20', INTERVAL '3 1:20' DAY_MINUTE)
```

Результат: 2015-10-01 11:50:20

Функция SUBDATE (date, INTERVAL value) производит вычитание значения value из датыскпс.

Пример: \_\_\_\_\_

```
SELECT SUBDATE ('2015-09-28 10:30:20', INTERVAL 20 HOUR)
```

Результат: 2015-09-27 14:30:20

Функция PERIODADD (period, n) прибавляет к значению period n месяцев. Значение период должно быть представлено в формате YYYYMM (например сентябрь 2015 года будет 201509), Пример: \_\_\_\_\_

```
SELECT PERIOD ADD (201509, 4)
```

Результат: 201601

Функция TIMESTAMP ADD (interval, n, date) прибавляет к дате date временной интервалл, значения которого задаются параметром interval. Возможные значения параметра interval:

- FRAC\_SECOND — микросекунды
- SECOND — секунды
- MINUTE — минуты
- HOUR — часы
- DAY — дни
- WEEK — недели
- MONTH — месяцы
- QUARTER — кварталы
- YEAR — годы

Пример:

```
SELECT TIMESTAMP ADD (QUARTER, 1, '2015-09-28')
```

Результат: 2015-12-28

Функция SUBTIME (date, time) вычитает из даты date время time. Пример:

Вычисление интервала между датами

Функция TIMEDIFF (date1, date2) вычисляет разницу в часах, минутах и секундах между двумя датами date1 и date2. Пример:

```
SELECT SUBTIME ('2015-09-28 10:30:20', '50:20:19')
```

```
SELECT TIMEDIFF ('2015-09-28 10:30:20', '2015-09-29 10:30:20')
```

Результат: -24:10:00

Функция DATEDIFF (date 1, date2) вычисляет разницу в днях между двумя датами, при этом часы, минуты и секунды при указании дат игнорируются. Пример:

```
SELECT DATEDIFF ('2015-09-28 00:00:20', '2015-09-27 23:40:20')
```

Результат: 1

С помощью этой функции легко определить сколько дней прошло с даты публикации статьи:

```
SELECT DATEDIFF (CURDATE (), date publication) FROM posts WHERE id_post = 1
```

Функция PERIODDIFF (period 1, period2) вычисляет разницу в месяцах между двумя датами. Даты должны быть представлены в формате YYYYMM. Например, узнаем сколько месяцев прошло с января 2015 по сентябрь 2015:

```
SELECT PERIODDIFF (201509, 201501)
```

Результат: 9

Функция TIMESTAMPDIFF (interval, date1, date2) вычисляет разницу между датами date2 и date1 в единицах указанных в параметре interval. При этом interval может принимать следующие значения:

- FRAC\_SECOND — микросекунды
- SECOND — секунды
- MINUTE — минуты
- HOUR — часы
- DAY — дни
- WEEK — недели
- MONTH — месяцы
- QUARTER — кварталы
- YEAR — годы

Пример:

```
SELECT TIMESTAMPDIFF (HOUR, '2015-09-28 10:30:20', '2015-09-28 19:50:20')
```

Результат: 9

Получение различных форматов даты и времени и другой информации \_\_\_\_\_ Функция DATE (datetime) возвращает дату, отсекая время. Пример: \_\_\_\_\_

```
SELECT DATE ('2015-09-28 10:30:20')
```

Результат: 2015-09-28

\_\_\_\_\_ Функция TIME (datetime) возвращает время, отсекая дату. Пример: \_\_\_\_\_

```
SELECT TIME ('2015-09-28 10:30:20')
```

Результат: 10:30:20

Функция **TIMESTAMP** (date) возвращает полный формат со временем даты date. Пример: \_\_\_\_

**TIMESTAMP** ('2015-09-28')

Результат: 2015-09-28 00:00:00

**DAY** (date) и **DAYOFMONTH** (date). Функции-синонимы, которые возвращают порядковый номер дня месяца. Пример: \_\_\_\_\_

**SELECT DAY** ('2015-09-28'), **DAYOFMONTH** ('2015-09-28')

Результат: 28 | 28

Функции **DAYNAME** (date), **DAYOFWEEK** (date) и **WEEKDAY** (date). Первая функция возвращает название дня недели, вторая — номер дня недели (отсчет от 1 — воскресенье до 7 — суббота), третья также номер дня недели только другой отсчет(отсчет от 0 — понедельник, до 6 — воскресенье). Пример: \_\_\_\_

**SELECT DAYNAME** ('2015-09-28'), **DAYOFWEEK** ('2015-09-28'), **WEEKDAY** ('2015-09-28') Результат:

Monday 2 | 0

Функции **WEEK** (date) и **WEEKOFYEAR** (datetime). Обе функции возвращают номер недели в году, только у первой неделя начинается с воскресенья, а у второй с понедельника. Пример: \_\_\_\_\_

**SELECT WEEK** ('2015-09-28 10:30:20'), **WEEKOFYEAR** ('2015-09-28 10:30:20')

Результат: 39 | 40

Функция **MONTH** (date) возвращает числовое значение месяца (от 1 до 12), а **MONTHNAME** (date) название месяца. Пример:

**SELECT MONTH** ('2015-09-28 10:30:20'), **MONTHNAME** ('2015-09-28 10:30:20')

Результат: 9 | September

Функция **QUARTER** (date) возвращает номер квартала года (от 1 до 4). Пример:

**SELECT QUARTER** ('2015-09-28 10:30:20')

Результат: 3

Функция **YEAR** (date) возвращает значение года (от 1000 до 9999). Пример:

**SELECT YEAR** ('2015-09-28 10:30:20')

Результат: 2015

Функция **DAYOFYEAR** (date) возвращает порядковый номер дня в году (от 1 до 366). Пример:

**SELECT DAYOFYEAR** ('2015-09-28 10:30:20')

Результат: 271

Функция **HOUR** (datetime) возвращает значение часа (от 0 до 23). Пример:

**SELECT HOUR** ('2015-09-28 10:30:20')

Результат: 10

Функция **MINUTE** (datetime) возвращает значение минут (от 0 до 59). Пример:

**SELECT MINUTE** ('2015-09-28 10:30:20')

Результат: 30

Функция **SECOND** (datetime) возвращает значение секунд (от 0 до 59). Пример:

Функция EXTRACT (type FROM date) возвращает часть даты date определяемую параметром type.

Пример: \_\_\_\_\_

```
SELECT EXTRACT (YEAR FROM '2015-09-28 10:30:20'), EXTRACT (MONTH FROM '2015-09-28 10:30:20'),  
EXTRACT (DAY FROM '2015-09-28 10:30:20'), EXTRACT (HOUR FROM '2015-09-28 10:30:20'), EXTRACT  
(MINUTE FROM '2015-09-28 10:30:20'), EXTRACT (SECOND FROM '2015-09-28 10:30:20')
```

Результат: 2015 | 9 | 28 | 10 | 30 | 20

Взаимобратные функции TO DAYS (date) и FROMDAYS (n). Первая преобразует дату в количество дней, прошедших с нулевого года. Вторая, наоборот, принимает число дней, прошедших с нулевого года и преобразует их в дату. Пример: \_\_\_\_\_

```
SELECT TO DAYS ('2015-09-28 10:30:20'), FROMDAYS (736234)
```

Результат: 736234 | 2015-09-28

Взаимобратные функции UNIXTIMESTAMP (date) и FROM UNIXTIME (n). Первая преобразует дату в количество секунд, прошедших с 1 января 1970 года. Вторая, наоборот, принимает число секунд, с 1 января 1970 года и преобразует их в дату. Пример:

```
SELECT UNIX_TIMESTAMP ('2015-09-28 10:30:20'), FROMUNIXTIME (1443425420)
```

Результат: 1443425420 | 2015-09-28 10:30:20

Взаимобратные функции TIMETOSEC (time) и SEC TO TIME (n). Первая преобразует время в количество секунд, прошедших от начала суток. Вторая, наоборот, принимает число секунд с начала суток и преобразует их во время. Пример: \_\_\_\_\_

```
SELECT TIME TO SEC ('10:30:20'), SEC_TO_TIME (37820)
```

Результат: 37820 | 10:30:20

Функция MAKEDATE (year, n) принимает год year и номер дня в году n и преобразует их в дату.

Пример: \_\_\_\_\_

```
SELECT MAKEDATE (2015, 271)
```

Результат: 2015-09-28