

Курс: «Microsoft SQL Server»

Практическая часть экзамена состоит из одного задания. Студент может выбрать любое из заданий, указанных ниже. Если студент в рамках практических или домашних заданий работал над похожей задачей он должен выбрать другую задачу. Обратите внимание, что в каждом задании перечисляются данные для хранения, а не структура базы данных. Если для решения поставленных задач вам необходимы дополнительные данные, добавляйте их.

Первый вариант:

Для туристического агентства «Вокруг света за 80 дней» необходимо создать базу данных. В базе данных нужно хранить такую информацию:

- Работники турагентства
 - ▷ ФИО
 - ▷ должность
 - ▷ контактный телефон и email
 - ▷ дата приёма на работу
- Сотрудники, ответственные за страны, туры
- Туры
 - ▷ название тура
 - ▷ стоимость
 - ▷ дата старта и окончания
 - ▷ способ(ы) передвижения
 - ▷ страны и города тура (хранить информацию о датах посещения)

- ▷ достопримечательности в каждой точке маршрута (включенные в стоимость тура и опциональные за дополнительную плату)
- ▷ изображения достопримечательностей
- ▷ информация о гостиницах, где будут ночевать туристы в туре
- ▷ изображения гостиниц
- ▷ список туристов, оплативших тур
- ▷ максимальное количество туристов для тура
- ▷ список потенциальных туристов, которые интересовались туром
- ▷ ФИО сотрудника, отвечающего за конкретный тур
- Клиенты агентства
 - ▷ ФИО
 - ▷ контактный телефон и email
 - ▷ дата рождения
 - ▷ будущие туры клиента
 - ▷ прошлые туры клиента
- Архив проведенных туров.

При проектировании базы данных обязательно используйте индексы. За отсутствие индексов или неправильное использование экзаменационная оценка может быть уменьшена.

Продумайте систему безопасности. Обязательные требования к ней:

- Пользователь с полным доступом ко всей информации (директор турагентства)

- Пользователи с доступом на чтение информации о странах и турах
- Пользователь с правом резервного копирования и восстановления данных
- Пользователь с правом создания и удаления пользователей
- Другие пользователи.

С помощью представлений, хранимых процедур, пользовательских функций, триггеров реализуйте следующую функциональность:

- Предоставьте информацию о всех актуальных турах
- Отобразите информацию о всех турах, которые стартуют в указанном диапазоне дат. Диапазон дат передаётся в качестве параметра
- Отобразите информацию о всех турах, которые посетят указанную страну. Страна передаётся в качестве параметра
- Отобразите самую популярную туристическую страну (по самому большому количеству туров с учетом архивных)
- Показать самый популярный актуальный тур (по максимальному количеству купленных туристических путевок)
- Показать самый популярный архивный тур (по максимальному количеству купленных туристических путевок)
- Показать самый непопулярный актуальный тур (по минимальному количеству купленных туристических путевок)
- Показать для конкретного туриста по ФИО список всех его туров. ФИО туриста передаётся в качестве параметра

- Проверить для конкретного туриста по ФИО находится ли он сейчас в туре. ФИО туриста передаётся в качестве параметра
- Отобразить информацию о том, где находится конкретный турист по ФИО. Если турист не в туре сгенерировать ошибку с описанием возникшей проблемы. ФИО туриста передаётся в качестве параметра
- Отобразить информацию о самом активном туристе (по количеству приобретённых туров)
- Отобразить информацию о всех турах указанного способа передвижения. Способ передвижения передаётся в качестве параметра
- При вставке нового клиента нужно проверять, нет ли его уже в базе данных. Если такой клиент есть, генерировать ошибку с описанием возникшей проблемы
- При удалении прошедших туров необходимо переносить их в архив туров
- Отобразить информацию о самой популярной гостинице среди туристов (по количеству туристов)
- При добавлении нового туриста в тур проверять не достигнуто ли уже максимальное количество. Если максимальное количество достигнуто, генерировать ошибку с информацией о возникшей проблеме
- Настроить создание резервных копий с периодичностью раз в день.

Второй вариант:

Для веб-проекта «Афиша событий» необходимо создать базу данных. В базе данных нужно хранить такую информацию:

- Категории событий
 - ▷ спектакль
 - ▷ концерт
 - ▷ выставка
 - ▷ цирк
 - ▷ спорт
 - ▷ семинары и тренинги
 - ▷ кино
 - ▷ юмор
 - ▷ вечеринки
 - ▷ детям
 - ▷ другое
- Событие
 - ▷ название события
 - ▷ дата проведения или диапазон дат
 - ▷ страна проведения
 - ▷ город проведения
 - ▷ место проведения
 - ▷ время проведения
 - ▷ категория события
 - ▷ описание события
 - ▷ возрастные ограничения
 - ▷ изображение для афиши события

- ▷ максимальное количество билетов
- ▷ количество приобретенных билетов
- Клиенты
 - ▷ ФИО клиента
 - ▷ контактный email клиента
 - ▷ дата рождения
 - ▷ приобретенные клиентом билеты
 - название события
 - цена билета
- Архив событий.

При проектировании базы данных обязательно используйте индексы. За отсутствие индексов или неправильное использование экзаменационная оценка может быть уменьшена.

Продумайте систему безопасности. Обязательные требования к ней:

- Пользователь с полным доступом ко всей информации
- Пользователь с правом только на чтение данных
- Пользователь с правом резервного копирования и восстановления данных
- Пользователь с правом создания и удаления пользователей.

С помощью представлений, хранимых процедур, пользовательских функций, триггеров реализуйте следующую функциональность:

- Отобразите все актуальные события на конкретную дату. Дата указывается в качестве параметра

- Отобразите все актуальные события из конкретной категории. Категория указывается в качестве параметра
- Отобразите все актуальные события со стопроцентной продажей билетов
- Отобразите топ-3 самых популярных актуальных событий (по количеству приобретенных билетов)
- Отобразите топ-3 самых популярных категорий событий (по количеству всех приобретенных билетов). Архив событий учитывается
- Отобразите самое популярное событие в конкретном городе. Город указывается в качестве параметра
- Покажите информацию о самом активном клиенте (по количеству купленных билетов)
- Покажите информацию о самой непопулярной категории (по количеству событий). Архив событий учитывается.
- Отобразите топ-3 набирающих популярность событий (по количеству проданных билетов за 5 дней)
- Покажите все события, которые пройдут сегодня в указанное время. Время передаётся в качестве параметра
- Покажите название городов, в которых сегодня пройдут события
- При вставке нового клиента нужно проверять, нет ли его уже в базе данных. Если такой клиент есть, генерировать ошибку с описанием возникшей проблемы
- При вставке нового события нужно проверять, нет ли его уже в базе данных. Если такое событие есть, генерировать ошибку с описанием возникшей проблемы

- При удалении прошедших событий необходимо их переносить в архив событий
- При попытке покупки билета проверять не достигнуто ли уже максимальное количество билетов. Если максимальное количество достигнуто, генерировать ошибку с информацией о возникшей проблеме
- При попытке покупки билета проверять возрастные ограничения. Если возрастное ограничение нарушено, генерировать ошибку с информацией о возникшей проблеме
- Настроить создание резервных копий с периодичностью раз в день.