Дипломная работа по курсу SQL-разработчик
«Разработка базы данных для оператора ТАКСИ»

СОДЕРЖАНИЕ:

Выполнил: Аргентов Сергей

Преподаватель: Авдеев Артём

г. Липецк 2022 год

1. Проверка оптимальности структуры базы данных

Для определения оптимальности структуры разрабатываемой базы данных – схема таблиц, предложенная в задании, проверялась на соответствие следующим 6 условиям:

1) Таблицы базы данных находятся в третьей нормальной форме, для которой должны выполняться:

а) требования для **Нулевой нормальной формы**:

- В таблицах нет порядковых номеров столбцов и строк.

  ПРВЕРЕНО: Порядковые номера в таблицах предложенной схемы отсутствуют.

б) требования для **Первой нормальной формы**:

- В таблицах нет дублирования строк(записей) данных.

  Данное требование будет реализовано в следующих заданиях визуальным контролем при вводе данных.

- В каждом из столбцов таблиц находятся данные только одного типа.

  Данное требование будет реализовано в следующих заданиях по умолчанию при создании таблиц средствами СУБД.

- В ячейках таблиц хранятся атомарные значения атрибутов.

  ПРОВЕРЕНО: Поля всех таблиц предусмотрены для атомарных значений сущностей предметной области.

  Справочно: В полях «name» таблиц «passenger» и «driver» для имени пассажира и водителя могут быть записаны разные комбинации Имени, Фамилии и даже Отчества. Однако, разница комбинаций для предметной области не существенна, так как имя обращения пассажир и водитель выбирают себе сами. Описание бизнес-процесса предметной области смотри на рис.1.1. Кроме того, хранение и обработка отдельных обязательных реквизитов фамилии, отчества и имени может привести к ужесточению требований по обеспечению безопасности персональных данных, что существенно удорожает техническую и организационную информационную инфраструктуру бизнес-процесса. Значит наиболее целесообразно для данного бизнес-процесса за атомарное значение принять – только имя, а в случаях ввода самим клиентом сочетаний фамилии, отчества, имени – введённое сочетание.

в) требования для **Второй нормальной формы**:

- Все таблицы имеют ключи.

  ПРОВЕРЕНО: Во всех таблицах есть несоставные первичные ключи, кроме таблицы currency2country, которая не имеет никакого первичного ключа. В указанную таблицу добавлено поле первичного ключа currency countru id (на рис.1.2 выделено красным). Поскольку данная таблица по сути является таблицей сопоставления, то наличие несоставного первичного ключа в ней не обязательно, но теоретически требуется и, кроме того, может пригодиться для технических операций (анализа отдельных записей).

- В таблицах с составным первичным ключом нет зависимости отдельных неключевых атрибутов от части составного первичного ключа.

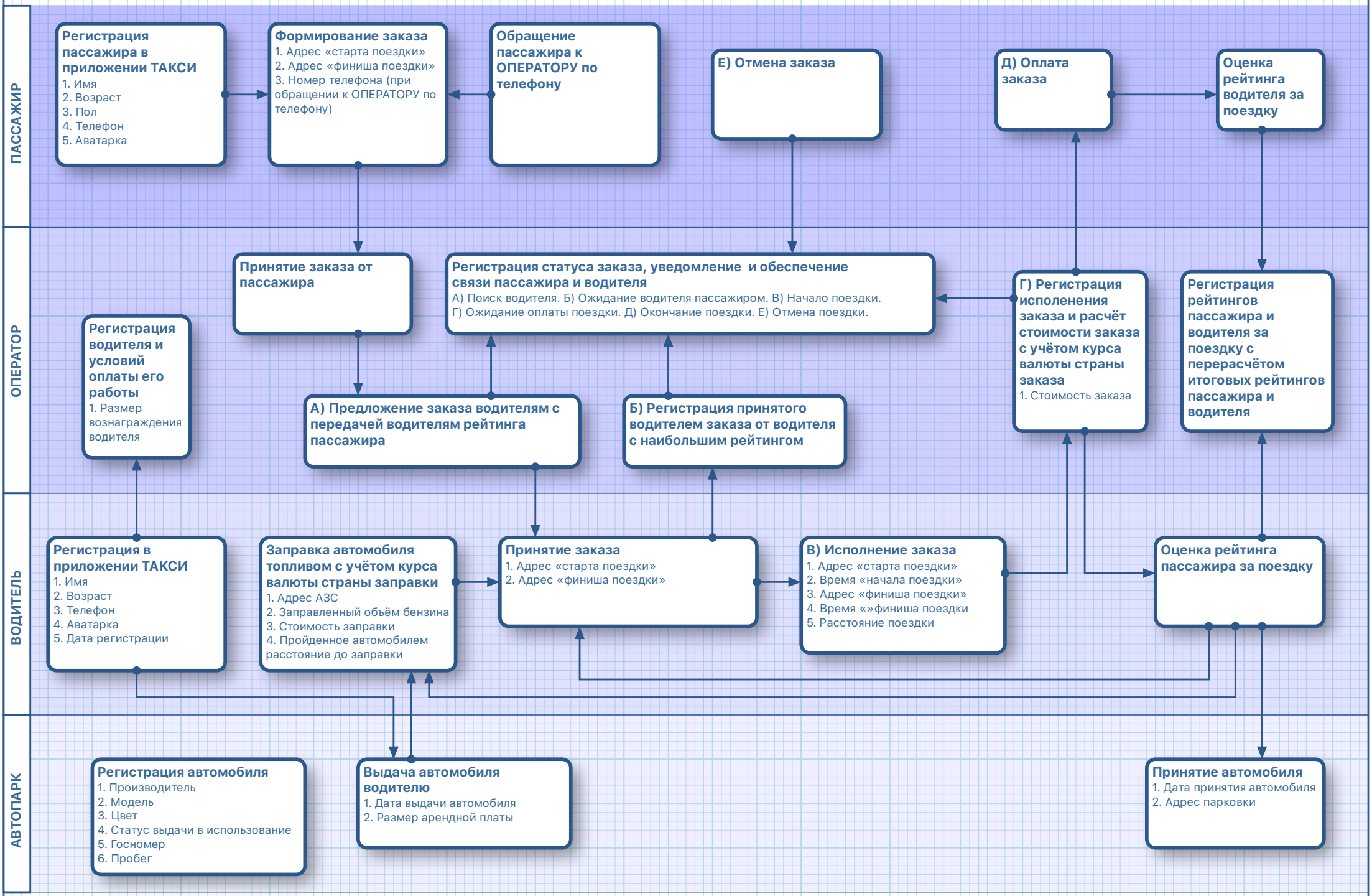
  ПРОВЕРЕНО: Составных первичных ключей в заданных таблицах нет.

г) требования для **Третьей нормальной формы**:

- В таблицах нет транзитивной зависимости неключевых атрибутов, то есть нет зависимости какого-либо неключевого атрибута от ключа через иной неключевой атрибут.

  ПРОВЕРЕНО: Транзитивной зависимости неключевых атрибутов в таблицах нет.

Рис.1.1. Бизнес-процесс ТАКСИ



2) В таблицах, указанных в задании схемы не должно быть: дублирующихся полей для одного и того же атрибута или совпадающих названий полей для разных атрибутов, а также полей для избыточных атрибутов, которые могут быть рассчитаны из других атрибутов, имеющихся в таблицах схемы.

ПРОВЕРЕНО: скорректированную по следующим позициям схему см. на рис.1.2.

а) В таблице «order» у двух полей одинаковое название «time_create». По-видимому, первое по порядку поле выделено под атрибут «время начала выполнения заказа», а второе под триггер, для которого предусмотрено поле в каждой таблице схемы. Соответственно для атрибута «время начала выполнения заказа» название поля заменено на «time_start».

б) В таблице «order» предусмотрено поле «average_driver_speed», по-видимому, для расчёта средней скорости движения во время заказа. Однако, средняя скорость является результатом деления пройденного пути (поле «distance» в таблице «way») на разницу времени окончания заказа (поле «time_end» в таблице «order») и времени начала заказа (поле «time_start» в таблице «order»). Соответственно поле «average_driver_speed» удалено из таблицы «order» так как в нём сохраняется производный атрибут, который может быть рассчитан программными средствами, а значит, в целях оптимизации пространства базы данных – не требующих постоянного хранения.

в) В таблице «rating_passenger2driver» предусмотрены два поля с название «time_create» для одного и того же атрибута «времени изменения данных в таблице». Соответственно одно из этих полей – удалено.

3) Часто используемые атрибуты сущностей вынесены в отдельные таблицы для предоставления возможности оптимизатору базы данных Oracle загружать их в оперативную память без других данных.

ПРОВЕРЕНО: В соответствие с бизнес-процессом работы ТАКСИ (см. рис.1.1) есть четыре атрибута, используемые чаще остальных (более одного раза за поездку):

а) Атрибут «рейтинг пассажира» используется водителем:
- при поиске водителя сразу после оформления заказа – водителям для согласия или отказа предоставляются сведения об общем рейтинге пассажира,
- при перерасчёте общего рейтинга пассажира после оценки пассажира водителем в конце поездки.

Данный атрибут уже вынесен в отдельную таблицу «passenger_rating» (см. рис.1.2).

б) Атрибут «рейтинг водителя» используется:
- при ответе водителей на запрос поиска водителя сразу после оформления заказа – в случае одновременного ответа нескольких водителей, приоритет отдаётся водителю с наибольшим рейтингом,
- при перерасчёте общего рейтинга водителя после оценки водителя пассажиром в конце поездки.

Данный атрибут уже вынесен в отдельную таблицу «passenger_rating» (см. рис.1.2).

в) Атрибуты «телефонный номер пассажира» и «телефонный номер водителя» используются несколько раз за поездку для уведомления о статусах пассажира и водителя, а также для обмена СМС сообщениями и совершения звонков между пассажиром и водителем.

Данные атрибуты не были вынесены в отдельные таблицы в задании, поэтому для них созданы отдельные таблицы «passenger_phone_number» и «driver_phone_number» (см. рис.1.2).

г) Атрибуты «статус заказа» используется несколько раз за поездку для уведомления участников о статусах.

Данный атрибут не был вынесен в отдельную таблицу в задании, поэтому для него создана отдельная таблица «status» (см. рис.1.2).

4) Большие по размеру атрибуты сущностей вынесены в отдельные таблицы для предоставления возможности оптимизатору Oracle загружать в оперативную память таблицы сущностей без загрузки таких атрибутов.

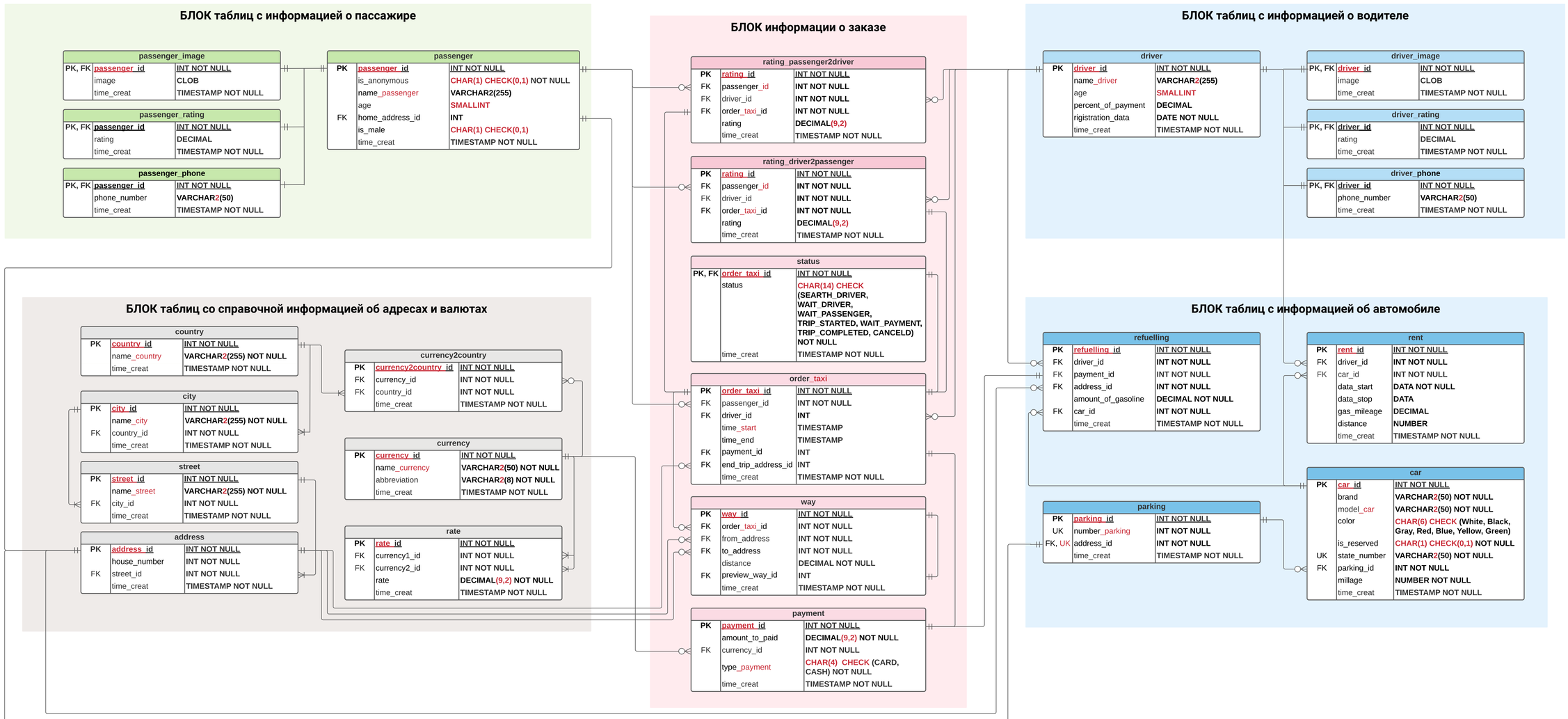
ПРОВЕРЕНО: Таких атрибутов два «passenger_image» и «driver_image», выделение которых в отдельные таблицы уже выполнено в схеме задания.

5) Первичные и внешние ключи в советующих таблиц поименованы идентично (для удобства написания запросов на объединение данных разных таблиц).

ПРОВЕРЕНО: Первичные ключи всех таблиц исправлены на идентичные названия с внешними ключами (см. рис.1.2 – скорректированные названия первичных ключей выделены красным).

Рис.1.2. Оптимизированная схема таблиц базы данных ТАКСИ

Аргентов Сергей 12.12.21 00:59:07



6) Кардинальность связей между таблицами соответствует предметной области.

ПРОВЕРЕНО: Кардинальность связей скорректирована (см. табл.1.1) с учётом бизнес-процесса предметной области (см. рис.1.1)

Таблица 1.1. Перечень скорректированных кардинальностей связей таблиц схемы

№	Связываемые ключи и таблицы (см.рис.1.2)	Кардинальность		Обоснование корректировки кардинальности относительно предложенного в схеме задания
		В задании	На рис.1.2	
1	PK.passanger FK.rating_passenger2driver и PK.driver FK.rating_passenger2driver	1..1 1..∞ 1..1 1..∞	1..1 0..∞ 1..1 0..∞	При регистрации пассажира в мобильном приложении ТАКСИ (до окончания первой поездки) может не быть ни одного рейтинга для водителя, выставленного этим пассажиром. Значит не будет строки таблицы «rating_passenger2driver» соответствующей пассажиру. Также если водитель недавно зарегистрировался и не совершил ни одной поездки – пассажир не проставит ему ни одного рейтинга.
2	PK.driver FK.rating_driver2passenger и PK.passanger FK.rating_driver2passenger	1..1 1..∞ 1..1 1..∞	1..1 0..∞ 1..1 0..∞	При регистрации водителя (до окончания первой поездки) может не быть ни одного рейтинга для пассажира, выставленного этим водителем. Значит не будет строки таблицы «rating_driver2passenger» соответствующей водителю. Также если пассажир недавно зарегистрировался и не совершившим ни одной поездки – водитель не проставит ему ни одного рейтинга.
3	PK.passanger FK.order_taxi	1..1 1..∞	1..1 0..∞	Перед началом первой поезди пассажира – заказа (order), может не быть ни одного, а значит строка таблицы «passenger» не будет соответствовать заказу в таблице «order». Это возможно в момент первой регистрации пассажира в мобильном приложении ТАКСИ.
4	PK.currency FK.payment	1..1 1..∞	1..1 0..∞	До завершения первой поезди по стране с определённой валютой – оплаты заказа (payment), может не быть ни одного, а значит ни одна строка таблицы «payment» не будет соответствовать имеющейся строке в таблице «currency».
5	PK.driver FK.order_taxi	1..1 1..∞	1..1 0..∞	Перед началом первой поезди водителя – заказа (order), может не быть ни одного, а значит ни одна строка таблицы «driver» не будет соответствовать заказу в таблице «order». Такое возможно в момент первой регистрации водителя в мобильном приложении ТАКСИ (до начала первой поездки).
6	PK.parking FK.car	1..1 1..∞	1..1 0..∞	Перед тем как в первый раз поставят автомобиль на новую, зарегистрированную в таблице «parking» парковку, эта парковка может не сопоставляться ни с одним автомобилем.
7	PK.currency FK.rate	… 1..∞	1…1 1..∞	По-видимому, в задании ошибочно пропущено указание на конкретную кардинальность связи со стороны PK.currency.
8	PK.currency FK.currency2country	… 1..∞	1…1 1..∞	По-видимому, в задании ошибочно пропущено указание на конкретную кардинальность связи со стороны PK.currency.
9	PK.car FK.refueling	… 1..∞	1…1 1..∞	По-видимому, в задании ошибочно пропущено указание на конкретную кардинальность связи со стороны PK.car.
10	PK.street FK.address	… 1..∞	1…1 1..∞	По-видимому, в задании ошибочно пропущено указание на конкретную кардинальность связи со стороны PK.street.
11	PK.city FK.street	… 1..∞	1…1 1..∞	По-видимому, в задании ошибочно пропущено указание на конкретную кардинальность связи со стороны PK.city.
12	PK.country FK.city	… 1..∞	1…1 1..∞	По-видимому, в задании ошибочно пропущено указание на конкретную кардинальность связи со стороны PK.country.

ДОПОЛНИТЕЛЬНО. Для дальнейшего удобства эксплуатации схемы таблиц (внесение изменений разработчиками, поиска ошибок типов данных, выбора сущностей/таблиц для дополнения атрибутов и т.п.) – схема была визуально разбита на 4 блока (подсвечены на рис.1.2 отдельными цветами):

- таблицы атрибутов ПАССАЖИРА (зелёный БЛОК);
- таблицы атрибутов ВОДИТЕЛЯ и АВТОМОБИЛЯ (синий БЛОК);
- таблицы атрибутов ЗАКАЗА (разовый БЛОК);
- таблицы атрибутов СПРАВОЧНОЙ ИНФОРМАЦИИ об адресах и валютах (серый БЛОК).

2. Разработка базы данных по оптимизированной схеме

2.1. Дополнительная коррекция атрибутов схемы таблиц, необходимость которой выявлена при создании таблиц средствами СУБД Oracle (см. на рис.1.2):

а) Поскольку в таблицах базы данных Oracle полю не может быть присвоен тип данных BOOLEAN (см. https://docs.oracle.com/en/database/oracle/oracle-database/21/lnpls/plsql-data-types.html#GUID-02AEA63C-9A27-43F4-80B7-4781343445D0), данный тип заменён в таблице passenger на CHAR(1) с выбираемым значениями (0,1). Тип VARCHAR2 в данном случае использовать менее целесообразно, так как для строк с фиксированной длинной (как в нашем случае), тип CHAR даёт лучшую производительность. Так же тип SHORT INTRGER в таблицах passenger и driver заменён на поддерживаемый базой данных тип SMALLINT.

б) Поскольку в таблицах базы данных Oracle полю не может быть присвоен тип данных ENUM в таблицах status, payment, car для полей с такими бизнес-требованиями был скорректирован тип данных. на CHAR с указанием макимального количества символов, которое требуется в соответствующем поле: для таблицы payment – CHAR(4); для таблицы status – CHAR(14); для таблицы car поля collor – CHAR(6); для таблицы car поля is_reserved – CHAR(1). К данным полям применено ограничение значения CHECK с указанием перечня возможных принимаемых значением поля.

в) Поскольку тип данных VARCHAR2 является стандартом Oracle, неизменность которого гарантирована и в нашей предметной области нет необходимости делать различие между хранением пустой строки и NULL (свойство типа VARCHAR), то в таблицах тип VARCHAR заменён на VARCHAR2.

г) Поскольку ключевое слово «order» зарезервировано в SQL – для исключения риска ошибок при компиляции кода название таблицы order заменено на order_taxi, а первичный ключ этой таблицы на скорректирован на order_taxi_id. Также скорректированы соответствующие внешние ключи в ключи в таблицах WAY, rating_passenger2driver, rating_driver2passenger.

д) Поскольку ключевое слово «name» зарезервировано в SQL – для исключения риска ошибок при компиляции кода названия соответствующих полей в таблицах passenger, driver, country, city, street, currency скорректированы на name_passenger, name_driver, name_country, name_city, name_street, name_currency.

е) Поскольку ключевое слово «number» зарезервировано в SQL – для исключения риска ошибок при компиляции кода название соответствующего поля в таблице parking скорректировано на number_parking.

ж) Поскольку ключевое слово «model» зарезервировано в SQL – для исключения риска ошибок при компиляции кода название соответствующего поля в таблице car скорректировано на model_car.

з) Поскольку ключевое слово «type» зарезервировано в SQL – для исключения риска ошибок при компиляции кода название соответствующего поля в таблице payment скорректировано на type_payment.

Основные корректировки схемы на рис.1.2. выделены красным шрифтом.

2.2. Создание схемы taxi_argentov (см. файл sys@xepdb1.sql).

Worksheet | Query Builder

```
create user taxi_argentov
    identified by taxi
    default tablespace users
    quota 100M on users
    temporary tablespace temp;

grant create session to taxi_argentov;

grant create table to taxi_argentov;
grant create procedure to taxi_argentov;
grant create trigger to taxi_argentov;
grant create view to taxi_argentov;
grant create sequence to taxi_argentov;

grant alter any table to taxi_argentov;
grant alter any procedure to taxi_argentov;
grant alter any trigger to taxi_argentov;

grant delete any table to taxi_argentov;

grant drop any table to taxi_argentov;
grant drop any procedure to taxi_argentov;
grant drop any trigger to taxi_argentov;
grant drop any view to taxi_argentov;

grant insert any table to taxi_argentov;

```

2.3. Создание таблиц по оптимизированной схеме, указанной на рис.1.2. и их заполнение тестовыми данными (см. файл TAXI.sql).

Схема создавалась последовательно по блокам (СПАВОЧНАЯ ИНФОРМАЦИЯ, ИНФОРМАЦИЯ О ПАССАЖИРЕ, ИНФОРМАЦИЯ О ВОДИТЕЛЕ И МАШИНЕ, ИНФОРМАЦИЯ О ЗАКАЗА). Такая последовательность создания таблиц обеспечила создание внешних ключей после создания советующих таблиц с первичными ключами.

Для разработки (перезаписи таблиц) в начале кода (см. файл TAXI.sql) был предусмотрен блок УДАЛЕНИЯ таблиц, в котором удаляемые таблицы располагались в обратном порядке их создания, что позволяло, при необходимости, удалять таблицы блоками: удаляя в блоке таблицы с внешними ключами перед удалением таблиц с соответствующими первичными ключами.

2.3. Разработка процедур внесения данных в таблицы (см. файл TAXI_procedure.sql).

В задание по разработке процедуры «ЗАПРАВКА АВТОМОБИЛЯ» внесена корректировка: для обеспечения практического использования процедуры для определения водителем адреса заправочной станции – добавлен входной параметр «адреса заправочной станции, на которой заправляется автомобиль: var_address_id».

2.4. Разработка конвейерной табличной функции для расчёта зарплаты водителей (см. файл TAXI_function.sql).

2.5. Разработка представлений для отчётов (см. файл TAXI_view.sql).

Для обеспечения возможности неограниченного во времени обновления курса валют, используемого в разных отчётах – разработаны запросы, выбирающие для каждой валюты наиболее релевантный по времени курс валют. То есть наиболее близкий «слева по оси времени» к времени параметра, определяемого отчётом и зависящего от соответствующего по времени курса валют (см. табл.2.5 и файл TAXI_view.sql)

Таблица 2.5. Запросы выбора наиболее релевантных по времени курсов валют

№	Группы запросов, участвующих в представлении	Представление
1	s_list_refuelling_and_all_earlier_exchange_rates s_list_refuelling_and_timely_rate	report_most_expensive_gasoline_prices
2	s_list_order_and_all_earlier_exchange_rates s_list_order_and_timely_rate	report_avg_country_order_prices
3	s_list_order_all_city_russia s_list_order_all_city_russia_rate	report_monthly_dynamics_fares

ДОПОЛНИТЕЛЬНО: по рекомендациям ОРАКЛ для повышения производительности запросов, используемых в представлениях и курсорах процедур и функций, все запросы выполнялись по принципам:

- выделение подзапросов в сложных комбинациях запросов;
- конфигурация подзапросов в запросе таким образом, чтобы при переходе от первого подзапроса к следующему подзапросу происходило снижение строк, обрабатываемых в каждом следующем подзапросе;
- минимизация повторяющихся объединений одних и тех-же таблиц.

3. Секционирование таблиц базы данных и индексирование полей таблиц базы данных

С целью повышения производительности работы базы данных при выполнении процедур, функций и представлений, выполняющих запросы к базе данных, выполнено секционирование таблиц базы данных и индексирование полей таблиц базы данных по следующим принципам:

- таблицы разбиваются на секции/подсекции по наиболее часто используемым в запросах полям при объединении таблиц оператором «JOIN» и фильтрации оператором «WHERE»;
- дополнительно к секционированию каждая секция/подсекция индексируется;
- при наличии в таблице более двух полей, которые используются при объединении таблиц оператором «JOIN» и фильтрации в запросах оператором «WHERE», на такие поля добавляется отдельный индекс.

Для подсчёта частоты использования в запросах полей отдельных таблиц (при реализации операторов «JOIN» и «WHERE») посчитано количество таких использований в процедурах, функциях и представлениях. Далее количество использований сгруппировано по полям и соответствующим таблицам (см. табл.3). Соответственно по количеству полей, по которым целесообразно провести оптимизацию секционированием и индексированием все таблицы разбиты на следующие группы:

а) Таблицы с одним используемым полем (позиции 1-3 табл.3) – разбиваются на секции. Поля не нуждаются в принудительной индексации, так как все они являются первичными ключами и на них уже создан индекс.

б) Таблицы с двумя используемыми полями (позиции 4-14 табл.3) – разбиваются на секции и подсекции. Для подсекций выбирается поле с наименьшим числом использования (см. колонку ВСЕГО в табл.3). Индексы принудительно проставляются на поля, не являющиеся первичными ключами:

rent.data_stop, currency.abbreviation, rating_driver2passeng.time_creat, rating_passenger2driver.time_creat, driver_rating.rating, rate.time_creat, passenger.is_anonymous.

в) Таблицы с тремя и более используемыми полями (позиции 15-18 табл.3) – разбиваются на секции и подсекции по аналогии с группой б). Индексы принудительно проставляются на поля, не являющиеся первичными ключами: payment.time_creat, country.name_country.

Метод секционирования, указанный для соответствующего поля (см. табл.3) выбран следующим образом:

для полей с датой – RANGE-метод с интервалом 1 месяц;
для полей с первичными ключами – HASH-метод с 4 секциями/подсекциями;
для полей с определённым списком значений – LIST-метод.

В файле TAXI_partition.sql представлен код секционирования таблиц и индексации при необходимости полей, используемых операторами «JOIN» и «WHERE».

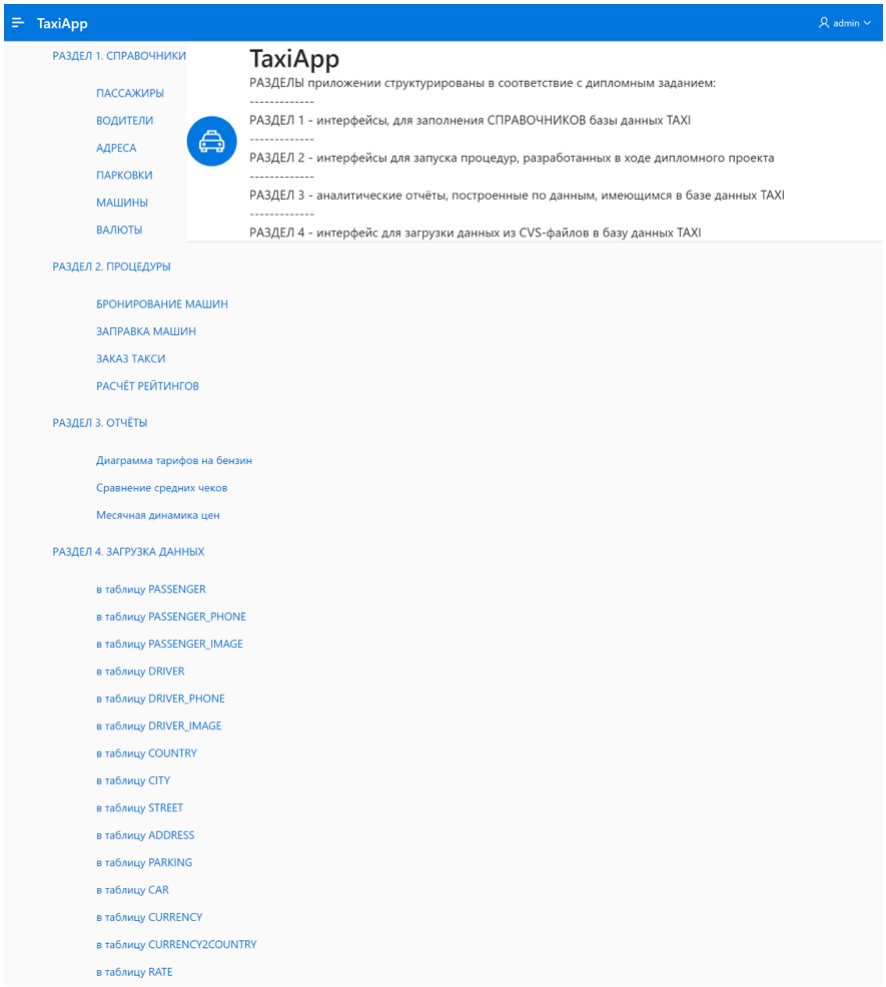
Таблица 3. Перечень полей таблиц, используемых в операторах «JOIN» и «WHERE»

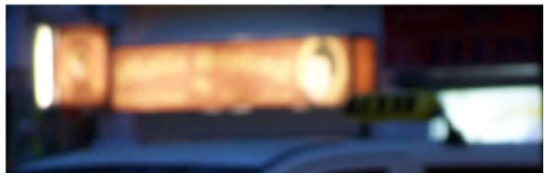
№	Таблица	Поле таблицы, частота использования которого в выражениях «JOIN» и «WHERE» определяется при подсчёте (ВЫБРАННЫЙ МЕТОД СЕКЦИОНИРОВАНИЯ)	ВСЕГО использований	Процедуры и функция							Представления				
				1. car_booking	2. car_unbooking	3. car_refuelling	4. create_order_taxi	5. calculation_rating_passenger	6. calculation_rating_driver	7. get_calculation_salary	8. report_new_drivers	9. report_frequently_visited_places	10. report_most_expensive_gasoline_prices	11. report_avg_country_order_prices	12. report_monthly_dynamics_fares
1	car	car_id (HASH)	2	1	1										
2	passenger_rating	passenger_id (HASH)	1					1							
3	driver	driver_id (HASH)	4							2	2				
4	rent	car_id (HASH)	2		1	1									
		data_stop (RANGE)	1			1									
5	currency	currency_id (HASH)	6										2	2	2
		abbreviation (LIST)	2			1	1								
6	rating_driver2passenger	passenger_id (HASH)	2					1	1						
		time_creat (RANGE)	1					1							
7	rating_passenger2driver	time_creat (RANGE)	2					1	1						
		driver_id (HASH)	1						1						
8	driver_rating	driver_id (HASH)	1						1						
		rating (HASH)	1								1				
9	rate	time_creat (RANGE)	8							2			2	2	2
		currency2_id (HASH)	4							1			1	1	1
10	way	order_taxi_id (HASH)	2									1			1
		preview_way_id (HASH)	1									1			
11	refuelling	payment_id (HASH)	2							1			1		
		driver_id (HASH)	1							1					
12	passenger	passenger_id (HASH)	3								2	1			
		is_anonymous (LIST)	2								2				
13	address	address_id (HASH)	4									1	1	1	1
		street_id (HASH)	4									1	1	1	1
14	street	street_id (HASH)	4									1	1	1	1
		city_id (HASH)	4									1	1	1	1
15	payment	time_creat (RANGE)	4							1			1	1	1
		currency_id (HASH)	4							1			1	1	1
		payment_id	3										1	1	1
16	city	city_id (HASH)	5									1	2	1	1
		country_id (HASH)	2									1			1
		street_id	1											1	
17	country	country_id (HASH)	3									1		1	1
		name_country (HASH)	1											1	
		time_creat	1												1
18	order_taxi	driver_id (HASH)	5							1	1	3			
		payment_id (HASH)	3							1				1	1
		passenger_id	1								1				
		order_taxi_id	1									1			
		end_trip_address_id	1												1

2. Разработка Apex приложения для просмотра и заполнения информацией базы данных

В соответствие с дипломным заданиям (включая его дополнительные части) разработано приложение с интерфейсами Apex, которые включают 4 раздела:

- ручное заполнение справочников,
- запуск разработанных процедур,
- отчёты по разработанным VIEW,
- заполнения справочников загрузкой данных из файлов CVS.





Все материалы по дипломному проекту, включая демонстрационное видео работы приложения Apex представлены по ссылке: https://disk.yandex.ru/d/C0sEr4zDxo-Qkg