

Дипломный проект по курсу SQL-разработчик «Разработка базы данных для сервиса ТАКСИ»

СОДЕРЖАНИЕ:

Слайд 2. О разработчике

Слайд 3-8. Шесть шагов оптимизации схемы таблиц базы данных

Слайд 9. Особенности разработанных запросов

Слайд 10. Методика определения правил секционирования таблиц

Слайд 11. Разработка демонстрационного приложения в Арех

Слайд 12. Темы для дальнейшего развития навыков pl/sql-разработки

Разработал: Аргентов Сергей
+7-910-742-76-24
argentovsg@mail.ru

Липецк – февраль 2022 г.

Преподаватель: Авдеев Артём

Аргентов Сергей



цели прохождения курса «SQL-разработчик»

повышение навыка оперативного доступа к данным Database ORACLE для ускорения принятия управленческих решений

получение дополнительной компетенции в сфере Database для участия в новых проектах с высоким информационно-технологическим потенциалом

... и ... просто интересно 😎

образование

- высшее: электротехническое, экономическое, юридическое
- аспирантура, кандидат технических наук

деятельность

20 лет в менеджменте ритейла энергетических компаний

проекты в компании с 600 тыс. клиентами

- конструирование логистики точек розничной сети (сокращение очередей в 3 раза)
- методическое обеспечение комплексной автоматизации ритейла (биллинг, CRM, BPM)
- конструирование интернет-сервиса для клиентов (увеличение пользователей в 60 раз)
- интеграция ИТ-систем в общие процессы ритейла (сокращение издержек на 25%)

3. ПРОВЕРЕНА ТРЕТЬЯ НОРМАЛЬНАЯ ФОРМА

НУЛЕВАЯ НОРМАЛЬНАЯ ФОРМА

нет порядковых номеров
строк и столбцов

ВТОРАЯ НОРМАЛЬНАЯ ФОРМА

добавлен первичный ключ
в таблицу `currency2country`

ПЕРВАЯ НОРМАЛЬНАЯ ФОРМА

нет дублирования строк, в
каждом столбце однотипные
данные – контролируется СУБД
при заполнении таблиц

все атрибуты в полях таблиц – атомарны
(в т.ч. ФИО для данной предметной области)

ТРЕТЬЯ НОРМАЛЬНАЯ ФОРМА

нет транзитивной
зависимости неключевого
атрибута от ключа через
другой неключевой атрибут

4. СКОРРЕКТИРОВАН НАБОР АТТРИБУТОВ В ТАБЛИЦАХ

ТАБЛИЦА «order_taxi»

исключен дублирующийся
атрибут «time_create»

добавлен атрибут «time_start» для
фиксации времени начала заказа,
которое необходимо фиксировать
для расчётов с пассажиром

исключен атрибут «average_driver_speed» как
избыточный для хранения в базе данных,
поскольку он вычисляется из других атрибутов

ТАБЛИЦА «rating_passenger2driver»

исключен дублирующийся
атрибут «time_create»

5. ДЕКОМПОЗИРОВАНЫ ТАБЛИЦЫ

Для возможности загрузки оптимизатором ORACLE в оперативную память минимальных объёмов таблиц, в отношении которых наиболее часто происходят операции чтения/записи данных (более 1-го раза при выполнении заказа) – в отдельные таблицы вынесены следующие атрибуты

номера телефонов – используются несколько раз для отправки смс о статусе заказа и общения водителя и пассажира

статус исполнения заказа – корректируется несколько раз в ходе исполнения заказа

6. УТОЧНЕНЫ ТИПЫ, НАЗВАНИЯ, УЛУЧШЕНО USABILITY

Типы данных полей
заменены на
допустимые в SQL

BOLLEAN	->	CHAR
SHOT INTEGER	->	SMALLINT
ENUM	->	CHAR + CHECK
VARCHAR	->	VARCHAR2

Переименованы поля, в
названиях которых были
зарезервированные слова

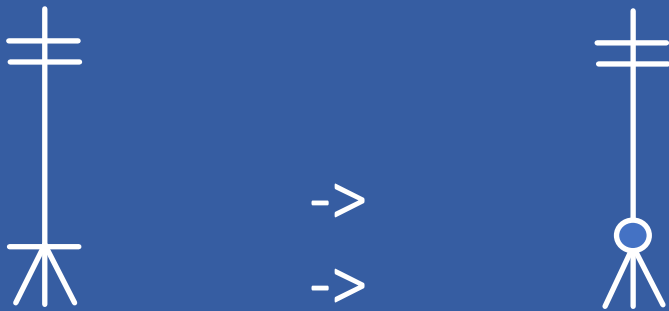
order	->	order_taxi
name	->	name_passenger, ...
number	->	number_parking
model	->	model_car
type	->	type_payment

Связанные первичные и внешние
ключи поименованы идентично для
удобства при написании запросов

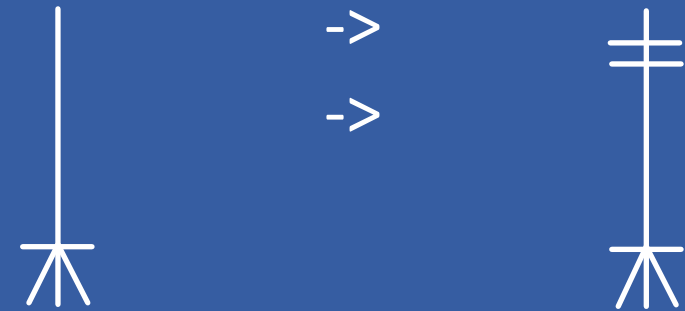
```
FROM s_list_cont_frequently_visited_places lcrfvp
LEFT OUTER JOIN passenger p
    ON lcrfvp.passenger_id = p.passenger_id
LEFT OUTER JOIN address a
    ON lcrfvp.address_id = a.address_id
LEFT OUTER JOIN street s
    ON a.street_id = s.street_id
LEFT OUTER JOIN city c
```

7. УТОЧНЕНЫ КАРДИНАЛЬНОСТИ СВЯЗЕЙ ТАБЛИЦ

в 8 связях кардинальность скорректирована, так как одной строке в родительской таблице может не соответствовать ни одна строка в дочерней таблице



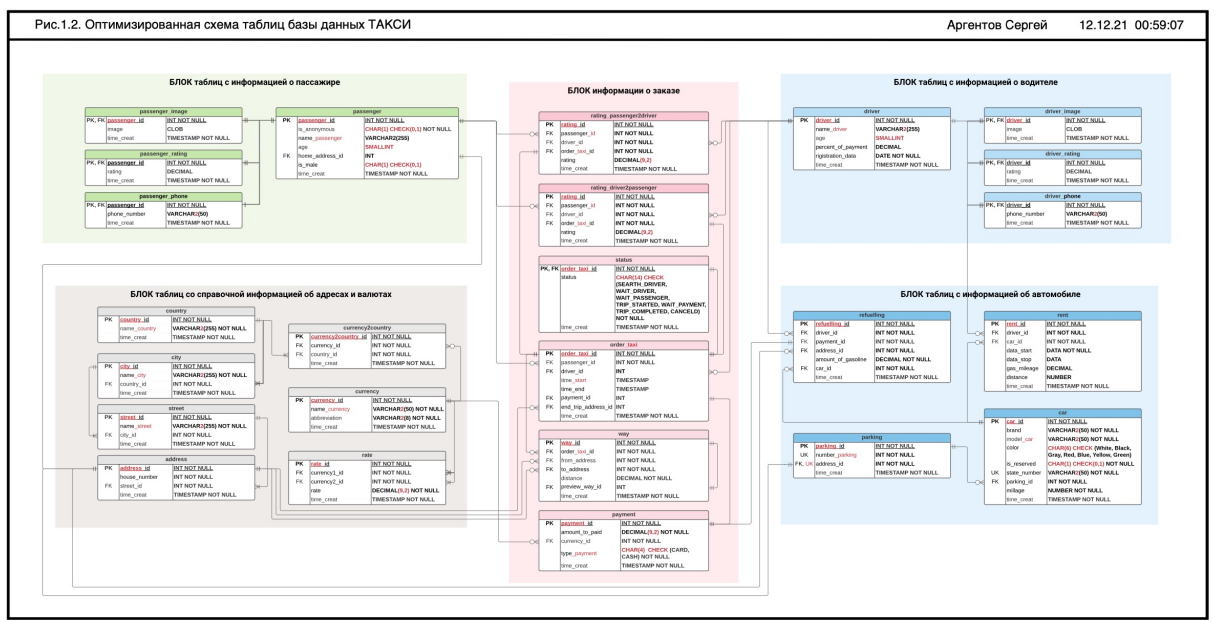
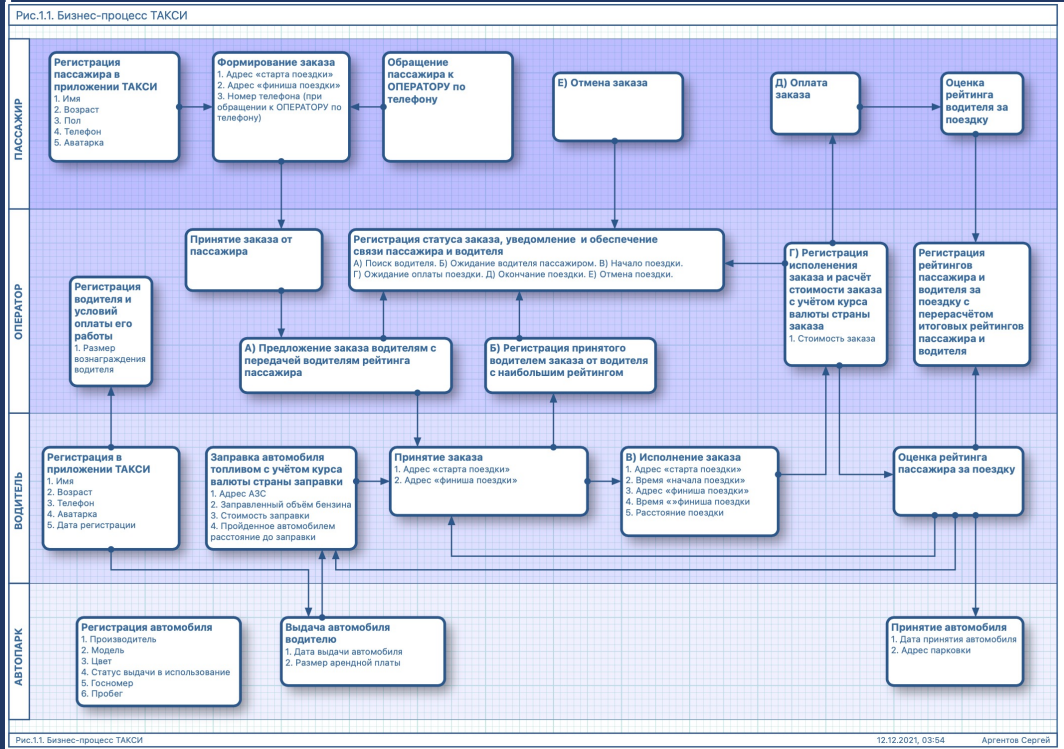
в 6 связях кардинальность добавлена со стороны родительской таблицы



8. УЛУЧШЕНО USABILITY ДОКУМЕНТАЦИИ ТАБЛИЦ

сформирована схема бизнес-процесса с указанием участников и выполняемых ими процедур

структурирована схема таблиц с выделением 4 блоков: справочники, пассажиры, водители-машины, заказы



оригиналы схем см. в пояснительной записке по ссылке: <https://disk.yandex.ru/d/C0sEr4Dxo-Qkg>

9. ОСОБЕННОСТИ РАЗРАБОТКИ ЗАПРОСОВ

РАЗРАБОТАННЫЕ ЗАПРОСЫ ОПТИМИЗИРОВАНЫ С УЧЁТОМ РЕКОМЕНДАЦИЙ ORACLE

конструкцией «WITH ... AS» все сложные запросы декомпозированы на подзапросы

от подзапроса к подзапросу снижается количество обрабатываемых строк

в запросах минимизированы повторы объединения одних и тех же таблиц

РАЗРАБОТАН АЛГОРИТМ ВЫБОРА РЕЛЕВАНТНОГО КУРСА ВАЛЮТ ПРИ ЛЮБОЙ ЧАСТОТЕ РЕГИСТРАЦИЙ КУРСОВ ВАЛЮТ

для каждого курса валют формируются массивы курсов валют, зарегистрированных до даты расчёта

из сформированных массивов выбираются курсы валют с максимальными датами регистрации

```
ksheet Query Builder
1 create or replace view report_most_expensive_gasoline_prices
2 as
3 WITH
4 -- Формируем список заправок автомобилей и курсов валют,
5 -- которые по времени были известны ранее заправки автомобиля
6 s_list_refuelling_and_all_earlier_exchange_rates AS
7 (SELECT rf.refuelling_id      as refuelling_id,
8      rf.address_id           as address_id,
9      rf.amount_of_gasoline   as amount_of_gasoline,
10     rf.payment_id            as payment_id,
11     p.amount_to_paid         as amount_to_paid,
12     p.currency_id            as currency_id,
13     r.time_creat             as rate_time
14 FROM   refuelling rf
15 LEFT OUTER JOIN payment p
16         ON rf.payment_id = p.payment_id
17 LEFT OUTER JOIN currency c
18         ON p.currency_id = c.currency_id
19 LEFT OUTER JOIN rate r
20         ON c.currency_id = r.currency2_id
21         and p.time_creat > r.time_creat),
22
23 -- Формируем список заправок автомобилей с подходящим каждой
24 -- заправке временем регистрации курса валют
25 s_list_refuelling_and_timely_rate AS
26 (SELECT refuelling_id, address_id, amount_of_gasoline,
27     payment_id, amount_to_paid, currency_id,
28     max(rate_time) as time_timely_rate
29 FROM   s_list_refuelling_and_all_earlier_exchange_rates
30 GROUP BY refuelling_id, address_id, amount_of_gasoline,
31     payment_id, amount_to_paid, currency_id),
```

10. СФОРМИРОВАНА МЕТОДИКА СЕКЦИОНИРОВАНИЯ

1. ОПРЕДЕЛЕНА ПОЛЯ СЕКЦИОНИРОВАНИЯ

все таблицы и поля, которые участвуют в операторах «WHERE» и «JOIN»

2. РАССЧИТАНА ЧАСТОТА ОБРАЩЕНИЙ К ПОЛЯМ

для каждого поля из п.1 определено количество участия в операторах «WHERE» и «JOIN»

3. СЕКЦИОНИРОВАНЫ ПОЛЯ С НАИБОЛЬШЕЙ ЧАСТОТОЙ

- в каждой таблице из п.1 создаются секции и подсекции на два поля с наибольшей частотой, определённой по п.3
- метод секционирования выбирается по типу поля: HASH, RANG, LIST

4. ОСТАВШИЕСЯ ПОЛЯ ИЗ п.1 – ПРОИНДЕКСИРОВАНЫ

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						
		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

расчёт см. в пояснительной записке по ссылке:
<https://disk.yandex.ru/d/C0sEr4zDxo-Qkg>

11. РАЗРАБОТКА ПРИЛОЖЕНИЯ В APEX

СТРУКТУРА РАЗДЕЛОВ ПРИЛОЖЕНИЯ СООТВЕТСТВУЕТ ЗАДАНИЮ

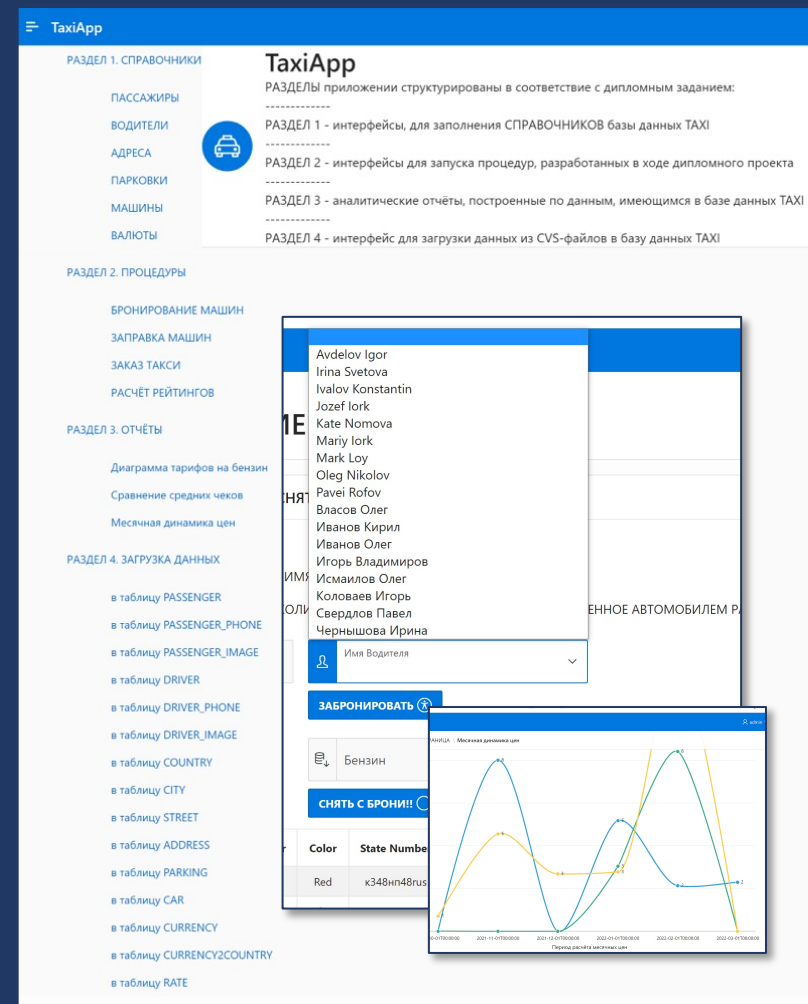
ручное заполнение справочников, интерфейсы процедур, интерфейсы отчётов, заполнение справочников из CVS-файлов

ИСПОЛЬЗОВАНЫ РАЗНООБРАЗНЫЕ ФОРМЫ ОТЧЁТОВ И ОПЦИЙ

Interactive Grid, Classic Report, Chart – с разными опциями: размер на экране, редактирование, пагинация, маска полей и т.п.

ИСПОЛЬЗОВАНЫ ОПЦИИ Shared Components

- **List Of Values** для визуализации списков выбора
- **Dynamic Actions** для инициации перерасчётов в интерфейсах
- **Navigation Menu** для оформления боковой навигации
- **Breadcrumbs** для оформления верхней навигации
- **Static Application Files** для загрузки логотипов



видео с демонстрацией приложения см. по ссылке:
<https://disk.yandex.ru/d/C0sEr4zDxo-Qkg>

12. ТЕМЫ ДЛЯ РАЗВИТИЯ НАВЫКОВ

1. ОРГАНИЗАЦИЯ РАЗНООБРАЗНЫХ КАРДИНАЛЬНОСТЕЙ СВЯЗЕЙ ТАБЛИЦ СРЕДСТВАМИ СУБД

Можно ли ограничениями на полях таблицы задать кардинальность $(1..1) \times (1..\infty)$?

2. ПРИМЕНЕНИЕ ХИНТОВ

По какой методологии подбирать хинты, где посмотреть их полный перечень и свойства?

3. ПРАКТИКА!!!

Где найти команду, в которой можно стажироваться в качестве начинающего разработчика pl/sql Oracle на небольшом объёме реальных работ?