

## Запросы DML:

№	Шаблон	Решение	Описание
1	SELECT col1, col2, ...colN FROM tableName;	<b>SELECT</b> airport_name, city, latitude, longitude, timezone <b>FROM</b> bookings.airports;	Вывод информации о аэропорте: название, город, координаты, временная зона.
2	SELECT DISTINCT col1, col2, ...colN FROM tableName;	<b>SELECT DISTINCT</b> arrival_airport <b>FROM</b> bookings.flights;	Вывод уникальных названий аэропортов из списка полётов.
3	SELECT col1, col2, ...colN FROM tableName WHERE condition;	<b>SELECT</b> model <b>FROM</b> bookings.aircrafts <b>WHERE</b> range = '3000'	Вывод модели самолёта, у которого дальность полётов равна 3000 км.
4	SELECT col1, col2, ...colN FROM tableName WHERE condition1 AND OR condition2;	<b>SELECT</b> airport_name, city <b>FROM</b> bookings.airports <b>WHERE</b> latitude > '51' <b>and</b> latitude < '77' <b>and</b> longitude > '58' <b>and</b> longitude < '180'	Вывод названий русских аэропортов и городов за уральскими горами.
5	SELECT col2, col2, ...colN FROM tableName WHERE colName IN (val1, val2, ...valN);	<b>SELECT</b> flight_id <b>FROM</b> bookings.flights <b>WHERE</b> departure_airport <b>IN</b> ( 'MJZ', 'NBC', 'NOZ', 'NAL', 'OGZ' );	Вывод идентификаторов полёта, аэропорт назначения которых в списке из пяти аэропортов.
6	SELECT col1, col2, ...colN FROM tableName WHERE colName BETWEEN val1 AND val2;	<b>SELECT</b> model <b>FROM</b> bookings.aircrafts <b>WHERE</b> range <b>BETWEEN</b> '2500' <b>AND</b> '6000'	Вывод моделей среднемагистральных самолётов (от 2500 до 6000 км).
7	SELECT col1, col2, ...colN FROM tableName WHERE colName LIKE pattern;	<b>SELECT</b> airport_code, city <b>FROM</b> bookings.airports <b>WHERE</b> city <b>LIKE</b> '%ск'	Вывод кодов аэропортов, у которых названия городов оканчиваются на «ск».
8	SELECT col1, col2, ...colN FROM tableName WHERE condition ORDER BY colName [ASC DESC];	<b>SELECT</b> ticket_no, amount, fare_conditions <b>FROM</b> bookings.ticket_flights <b>WHERE</b> fare_conditions = 'Business' <b>OR</b> fare_conditions = 'Comfort' <b>ORDER BY</b> amount <b>DESC</b>	Вывод номера билета, цены и класса обслуживания самых дорогих билетов. При этом класс обслуживания комфорт или бизнес-класс.
9	SELECT SUM(colName) FROM tableName WHERE condition GROUP BY colName;	<b>SELECT</b> SUM(amount) as Sum_Of_Amount <b>FROM</b> bookings.ticket_flights <b>WHERE</b> flight_id = '30625' <b>GROUP BY</b> fare_conditions	Вывод суммы стоимости билетов по классам обслуживания на рейсе 30625.
10	SELECT COUNT(colName) FROM tableName WHERE condition;	<b>SELECT</b> COUNT(ticket_no) as count_tickets <b>FROM</b> bookings.ticket_flights <b>WHERE</b> flight_id = '30628';	Вывод количества билетов на рейсе 30628.
11	SELECT SUM(colName) FROM tableName WHERE condition GROUP BY colName HAVING (function condition);	<b>SELECT</b> AVG(amount) <b>AS</b> average_price <b>FROM</b> bookings.ticket_flights <b>WHERE</b> fare_conditions <b>IN</b> ( 'Comfort', 'Business' ) <b>AND</b> flight_id = '30630' <b>GROUP BY</b> fare_conditions <b>HAVING</b> COUNT(ticket_no) > 10	Вывод средней стоимости билетов комфорт и бизнес- класса на рейсе 30630 с группировкой по классу обслуживания при количестве билетов на рейс более 10.

## Запросы DDL:

№	Шаблон	Решение	Описание
1	CREATE TABLE tableName ( col1 datatype, col2 datatype, ... colN datatype, PRIMARY KEY (одна или более колонка) );	<b>CREATE TABLE</b> fare_condition( fare_id integer <b>NOT</b> <b>NULL</b> , fare_name character(50) <b>NOT NULL</b> , <b>CONSTRAINT</b> fare_pkey <b>PRIMARY KEY</b> (fare_id) )	Создание таблицы для хранения значений классов обслуживания в соответствии с правилами приведения к НФ. Определение первичного ключа и колонки хранения названия класса обслуживания.
2	DROP TABLE tableName;	<b>DROP TABLE</b> fare_condition;	Удаление созданной таблицы.
3	CREATE UNIQUE INDEX indexName ON tableName (col1, col2, ...colN);	<b>CREATE UNIQUE INDEX</b> index_fare <b>ON</b> fare_condition (fare_id);	Создание индекса для таблицы.
4	ALTER TABLE tableName DROP INDEX indexName;	<b>DROP INDEX</b> index_fare	Удаление индекса у таблицы.
5	DESC tableName;	\d bookings.aircrafts	Аналог операции DESC через командную строку psql.
6	TRUNCATE TABLE tableName;	<b>TRUNCATE TABLE</b> fare_condition;	Очистка таблицы.
7	ALTER TABLE tableName ADD DROP MODIFY colName [datatype];	<b>ALTER TABLE</b> bookings.aircrafts <b>ADD</b> capacity integer	Добавление столбца вместимость самолета в таблицу хранения данных о самолётах.
8	ALTER TABLE tableName RENAME TO newTableName;	<b>ALTER TABLE</b> fare_condition <b>RENAME TO</b> fare;	Изменение имени таблицы.
9	INSERT INTO tableName (col1, col2, ...colN) VALUES (val1, val2, ...valN)	<b>INSERT INTO</b> fare_condition <b>VALUES</b> (1, 'Economy'), (2, 'Comfort'), (3, 'Business')	Вставка значений в таблицу.
10	UPDATE tableName SET col1 = val1, col2 = val2, ...colN = valN [WHERE condition];	<b>UPDATE</b> fare_condition <b>SET</b> fare_id = '1', fare_name = 'Economy' <b>WHERE</b> fare_id = '0';	Обновление значений таблицы записи с идентификатором 0 на запись с идентификатором 1.
11	DELETE FROM tableName WHERE condition;	<b>DELETE FROM</b> fare_condition <b>WHERE</b> fare_id = '1';	Удаление записей с идентификатором 1.