HW2

1) запуск докера `docker compose up -d`
2) Создаем директорию для БД:
    - `docker exec -it hive-server mkdir -p /data/warehouse/analytics`
3) Подключение к Hive по SQL: (для формирования скринов в отчете было использовано подключение через DBeaver)
    - `docker exec -it hive-server beeline`
    - `!connect jdbc:hive2://localhost:10000`
4) Создание базы данных:

```
CREATE DATABASE analytics
LOCATION '/data/warehouse/analytics';
```

5) Заполняем таблицы:

- из `airports.csv`

```
USE analytics;

CREATE EXTERNAL TABLE airports (
  airport_id INT,
  city STRING,
  state STRING,
  name STRING
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
  "separatorChar" = ",",
  "quoteChar" = "\"",
  "skip.header.line.count" = "1"
)
STORED AS TEXTFILE
LOCATION '/data/airports';

SELECT *
FROM airports
LIMIT 10;
```

- из `flights.csv`

```
CREATE EXTERNAL TABLE flights (
  day_of_month INT,
  day_of_week INT,
  carrier STRING,
  origin_airport_id INT,
  dest_airport_id INT,
```

	day_of_month	day_of_week	carrier	origin_airport_id	dest_airport_id	dep_delay	arr_delay
1	19	5	DL	11433	13303	-3	1
2	19	5	DL	14869	12478	0	-8
3	19	5	DL	14057	14869	-4	-15
4	19	5	DL	15016	11433	28	24
5	19	5	DL	11193	12892	-6	-11
6	19	5	DL	10397	15016	-1	-19
7	19	5	DL	15016	10397	0	-1
8	19	5	DL	10397	14869	15	24
9	19	5	DL	10397	10423	33	34
10	19	5	DL	11278	10397	323	322

Figure 1: ER diagram

```
  dep_delay INT,
  arr_delay INT
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
  "separatorChar" = ",",
  "quoteChar" = "\"",
  "skip.header.line.count" = "1"
)
STORED AS TEXTFILE
LOCATION '/data/flights';

SELECT *
FROM flights
LIMIT 10;
```

	day_of_month	day_of_week	carrier	origin_airport_id	dest_airport_id	dep_delay	arr_delay
1	19	5	DL	11433	13303	-3	1
2	19	5	DL	14869	12478	0	-8
3	19	5	DL	14057	14869	-4	-15
4	19	5	DL	15016	11433	28	24
5	19	5	DL	11193	12892	-6	-11
6	19	5	DL	10397	15016	-1	-19
7	19	5	DL	15016	10397	0	-1
8	19	5	DL	10397	14869	15	24
9	19	5	DL	10397	10423	33	34
10	19	5	DL	11278	10397	323	322

Figure 2: ER diagram

Витрины

1) Доля рейсов с задержкой по авиакомпаниям `carrier_delay_ratio`

```
CREATE VIEW carrier_delay_ratio AS
SELECT
  carrier,
  COUNT(*) AS total_flights,
```

```
  ROUND(SUM(CASE WHEN dep_delay > 0 THEN 1.0 ELSE 0.0 END) / COUNT(*),3) AS delay_ratio
FROM flights
WHERE dep_delay IS NOT NULL
GROUP BY carrier;

SELECT *
FROM carrier_delay_ratio
ORDER BY delay_ratio DESC, early_dep_ratio DESC;
```

	A-Z carrier	123 total_flights	123 delay_ratio
1	WN	575,739	0.594
2	F9	35,738	0.472
3	UA	286,418	0.465
4	FL	92,702	0.449
5	AA	289,855	0.436
6	VX	34,739	0.408
7	MQ	113,212	0.401
8	B6	121,906	0.381
9	EV	157,928	0.377
10	OO	160,164	0.315
11	YV	52,821	0.314
12	DL	381,657	0.313
13	US	233,321	0.294
14	9E	80,031	0.292
15	HA	17,432	0.239
16	AS	68,555	0.224

Figure 3: ER diagram

Описание: - Витрина показывает, какая доля рейсов каждой авиакомпании выполняется с задержкой вылета; - Задержкой считается рейс, у которого значение dep_delay больше нуля; - Используется относительная метрика, что позволяет корректно сравнивать авиакомпании разного масштаба; - Витрина позволяет выявить наименее и наиболее пунктуальные авиакомпании; - Полученные значения отражают вероятность задержки рейса для конкретной компании;

2) Средняя задержка вылета по дням недели `avg_delay_by_weekday`

```
CREATE VIEW avg_delay_by_weekday AS
SELECT
```

```
  day_of_week,
  ROUND(AVG(CASE WHEN dep_delay > 0 THEN dep_delay END),1) AS avg_delay_minutes,
  ROUND(AVG(CASE WHEN dep_delay < 0 THEN dep_delay END),1) AS avg_early_departure_minutes
FROM flights
WHERE dep_delay IS NOT NULL
GROUP BY day_of_week;

SELECT *
FROM avg_delay_by_weekday
ORDER BY day_of_week;
```

	day_of_week	avg_delay_minutes	avg_early_departure_minutes
1	1	31	-4.5
2	2	29.1	-4.7
3	3	31.9	-4.6
4	4	34.3	-4.5
5	5	32.1	-4.4
6	6	26.7	-4.7
7	7	29.4	-4.6

Figure 4: ER diagram

Описание: - Витрина показывает зависимость задержка вылета и раннего вылета от дня недели; - Для расчёта средней задержки учитываются только рейсы с положительным значением dep_delay; - Для расчёта раннего вылета учитываются только рейсы с отрицательным значением dep_delay; - Рейсы, вылетевшие точно по расписанию, не участвуют в расчётах; - Разделение задержек и ранних вылетов позволяет избежать взаимного сглаживания метрик;

3) Стабильность аэропортов (variance задержек) `airport_delay_stability`

```
CREATE VIEW airport_delay_stability AS
SELECT
  a.airport_id,
  a.name AS airport_name,
  ROUND(AVG(f.dep_delay),2) AS avg_delay,
  ROUND(STDDEV(f.dep_delay),2) AS std_delay
FROM flights f
JOIN airports a
  ON f.origin_airport_id = a.airport_id
WHERE f.dep_delay IS NOT NULL
GROUP BY a.airport_id, a.name
HAVING COUNT(*) > 50;

SELECT *
FROM airport_delay_stability
ORDER BY std_delay DESC;
```

Описание: - Витрина показывает среднюю задержку вылета и степень

	airport_id	airport_name	avg_delay	std_delay
1	14730	Louisville International-Standiford Field	11.15	43.98
2	11066	Port Columbus International	10.8	43.62
3	12953	LaGuardia	11.15	42.53
4	14027	Palm Beach International	11.52	42.25
5	12478	John F. Kennedy International	13.54	41.4
6	11618	Newark Liberty International	14.55	41.37
7	11193	Cincinnati/Northern Kentucky International	9.4	41.36
8	12264	Washington Dulles International	13.01	41.09
9	11278	Ronald Reagan Washington National	8.23	40.93
10	14524	Richmond International	8.87	40.84
11	12173	Honolulu International	5.65	40.56
12	13851	Will Rogers World	9.55	40.45
13	13931	Norfolk International	10.25	40.36
14	10792	Buffalo Niagara International	8.81	40.16
15	14492	Raleigh-Durham International	9.27	40.03
16	13930	Chicago O'Hare International	15.68	40.03

Figure 5: ER diagram

вариативности задержек для каждого аэропорта; - Средняя задержка отражает типичное отклонение времени вылета от расписания; - Стандартное отклонение характеризует стабильность работы аэропорта: чем выше значение, тем менее предсказуемы задержки; - Витрина позволяет выявить аэропорты с высоким уровнем нестабильности, даже при умеренной средней задержке;

4) Эффективность маршрутов (задержка на рейс) `route_efficiency`

```
CREATE VIEW route_efficiency AS
SELECT
  a1.name AS origin_airport,
  a2.name AS destination_airport,
  ROUND(AVG(f.dep_delay),2) AS avg_dep_delay
FROM flights f
JOIN airports a1 ON f.origin_airport_id = a1.airport_id
JOIN airports a2 ON f.dest_airport_id = a2.airport_id
WHERE f.dep_delay IS NOT NULL
GROUP BY a1.name, a2.name
HAVING COUNT(*) > 30;
```

```
SELECT *
FROM route_efficiency
ORDER BY avg_dep_delay DESC;
```

Описание:

- Витрина показывает среднюю задержку вылета для каждого маршрута -> ;
- Метрика avg_dep_delay отражает типичную задержку рейсов на конкретном направлении;

- Для исключения нестандартных направлений используются только маршруты с количеством рейсов более 30;
- Витрина позволяет выявить маршруты с систематически задержками;

	origin_airport	destination_airport	avg_dep_delay
1	Seattle/Tacoma International	Miami International	37.77
2	Chicago Midway International	Ontario International	32.8
3	Fort Lauderdale-Hollywood International	Richmond International	32.64
4	Chicago Midway International	San Francisco International	31.82
5	Norfolk International	Minneapolis-St Paul International	31.76
6	Metropolitan Oakland International	Logan International	31.54
7	Logan International	Metropolitan Oakland International	30.9
8	William P Hobby	LaGuardia	30.11
9	Lambert-St. Louis International	San Francisco International	29.58
10	John F. Kennedy International	Cincinnati/Northern Kentucky International	29.51
11	Dallas/Fort Worth International	Kahului Airport	29.31
12	Chicago Midway International	Jacksonville International	29.07
13	Newark Liberty International	Will Rogers World	29

Figure 6: ER diagram

5) Относительная нагрузка аэропортов `airport_traffic_share`

```
CREATE VIEW airport_traffic_share AS
SELECT
  airport_id,
  total_flights,
  ROUND(total_flights * 100.0 / SUM(total_flights) OVER (),2) AS traffic_share_prcnt
FROM (
  SELECT origin_airport_id AS airport_id, COUNT(*) AS total_flights
  FROM flights
  GROUP BY origin_airport_id

  UNION ALL

  SELECT dest_airport_id AS airport_id, COUNT(*) AS total_flights
  FROM flights
  GROUP BY dest_airport_id
) t;
```

```
SELECT *
FROM airport_traffic_share
ORDER BY traffic_share_prcnt DESC;
```

Описание: - Витрина показывает относительную долю трафика каждого аэропорта в общем количестве рейсов; - В расчёт включаются как рейсы на вылет, так и рейсы на прилёт; - Метрика traffic_share_prcnt выражена в процентах; - Витрина позволяет выявить наиболее крупные транспортные узлы и аэропорты с наибольшей нагрузкой;

6) Рейтинг авиакомпаний по пунктуальности `carrier_punctuality_rank`

	airport_id	total_flights	traffic_share_prcnt
1	10397	148,524	2.75
2	10397	148,563	2.75
3	13930	127,341	2.36
4	13930	127,195	2.35
5	12892	118,274	2.19
6	12892	117,714	2.18
7	11298	104,270	1.93
8	11298	103,939	1.92
9	11292	97,259	1.8
10	11292	96,919	1.79
11	14107	89,720	1.66
12	14107	89,814	1.66
13	14771	84,276	1.56
14	14771	84,063	1.56
15	12889	77,810	1.44
16	12889	77,878	1.44
17	11057	76,533	1.42
18	11057	76,465	1.41
19	12266	73,346	1.36

Figure 7: ER diagram

```
CREATE VIEW carrier_punctuality_rank AS
SELECT
  carrier,
  avg_delay,
  RANK() OVER (ORDER BY avg_delay ASC) AS punctuality_rank
FROM (
  SELECT
    carrier,
    AVG(dep_delay) AS avg_delay
  FROM flights
  WHERE dep_delay IS NOT NULL
  GROUP BY carrier
) t;
```

```
SELECT *
FROM carrier_punctuality_rank;
```

	A-Z carrier	123 avg_delay	123 punctuality_rank
1	AS	0.6592371089	1
2	HA	1.5339031666	2
3	US	4.9743315004	3
4	DL	7.4394836201	4
5	OO	7.8269398866	5
6	YV	9.3857556654	6
7	9E	9.5101898015	7
8	FL	10.1628875321	8
9	AA	12.0077970019	9
10	F9	12.1234540265	10
11	UA	12.5453882088	11
12	B6	12.6197972208	12
13	WN	12.8461664053	13
14	EV	14.1375373588	14
15	VX	14.3862517631	15
16	MQ	15.0501978589	16

Figure 8: ER diagram

Описание: - Витрина формирует рейтинг авиакомпаний на основе средней задержки вылета; - Средняя задержка рассчитывается как среднее значение по всем рейсам авиакомпании; - Авиакомпании сортируются по возрастанию средней задержки: чем меньше значение, тем выше пунктуальность;