17.03, ДОМАШНЯЯ РАБОТА

1) Номера деталей, поставляемых для любого проекта, разрабатываемого в Лондоне.

((j JOIN spj) (WHERE city = ‘London’)) [pn]

SELECT DISTINCT spj.pn FROM spj

JOIN j ON spj.jn = j.jn

WHERE city = 'London';

2) Определить номера проектов, для которых среднее количество поставляемых деталей с номером P1 больше, чем наибольшее количество любых деталей, поставляемых для проекта с номером J1.

// parts\_1 – находим номера проектов и среднее количество деталей среди всех поставок детали pn = 1 для данного проекта

// parts\_2 – определяем наибольшее количество деталей среди всех поставок для проекта jn = 1

// определяем исходные номера проектов

WITH (SUMMARIZE ((spj WHERE pn = 1) [jn, qty])

BY (jn) ADD AVG(qty) AS avg\_qty) AS parts\_1,

(MAX((spj WHERE jn = 1) [qty]) AS max\_qty) AS parts\_2:

((parts\_1 TIMES parts\_2) WHERE avg\_qty > max\_qty) [jn]

WITH parts\_1 AS (SELECT jn, AVG(qty) AS avg\_qty FROM spj WHERE pn = 1 GROUP BY jn),

parts\_2 AS (SELECT MAX(qty) AS max\_qty FROM spj WHERE jn = 1)

SELECT jn FROM parts\_1, parts\_2 WHERE avg\_qty > max\_qty;

3) Найти номера проектов, для которых поставщиками из Лондона не поставляется ни одна деталь красного цвета.

(j [jn])

MINUS

(((spj JOIN s JOIN (p [pn, color])) WHERE (city = 'London' AND color = 'Red')) [jn])

SELECT jn FROM j

EXCEPT

SELECT DISTINCT spj.jn FROM spj

JOIN s ON spj.sn = s.sn

JOIN p ON spj.pn = p.pn

WHERE (s.city = 'London' AND p.color = 'Red');

17.03, САМОСТОЯТЕЛЬНАЯ РАБОТА

ВАРИАНТ 1

1) Найти все номера проектов, детали для которых поставляются по крайней мере одним поставщиком из другого города.

((spj JOIN j

JOIN ((s RENAME city AS s\_city) [sn, s\_sity]))

WHERE city != s\_city) [jn]

SELECT DISTINCT spj.jn FROM spj

JOIN s ON spj.sn = s.sn

JOIN j ON spj.jn = j.jn

WHERE s.city <> j.city;

2) Определить общее количество деталей с номером PN в каждом из проектов, в которых участвует поставщик с заданным номером SN.

// pn = 1, sn = 1

// projects\_1 – выбираем номера проектов, в которых участвует поставщик sn = 1

// projects\_2 – собираем записи о поставках деталей pn = 1 во все проекты projects\_1

// определяем суммарное количество деталей для каждого проекта

WITH (((spj RENAME jn AS s\_jn) WHERE sn = 1) [s\_jn]) AS projects\_1,

((projects\_1 TIMES spj) WHERE (jn = s\_jn AND pn = 1) [jn, qty]) AS projects\_2:

projects\_2 GROUP [jn SUM(qty)] AS qty\_sum

WITH projects\_1 AS (SELECT DISTINCT jn AS s\_jn FROM spj WHERE sn = 1),

projects\_2 AS (SELECT spj.jn, spj.qty FROM spj, projects\_1

WHERE (jn = s\_jn AND pn = 1))

SELECT jn, SUM(qty) FROM projects\_2 GROUP BY jn;

ВАРИАНТ 2

1) Определить номера деталей, поставляемых для проектов поставщиком из того же города, в котором разрабатывается проект.

((spj JOIN (s RENAME city AS s\_city) JOIN (j RENAME city AS j\_city))

WHERE s\_city = j\_city) [pn]

SELECT DISTINCT spj.pn FROM spj

JOIN s ON s.sn = spj.sn

JOIN j ON j.jn = spj.jn

WHERE j.city = s.city;

2) Определить общее количество проектов, для каждой детали от поставщика с заданным номером SN.

// sn = 5

// parts\_1 – выбираем номера деталей, поставляемых поставщиком sn = 5

// parts\_2 – подсчитываем количество проектов для каждой детали

// выбираем детали parts\_1 и количество проектов для каждой из них

WITH (((spj RENAME pn AS s\_pn) WHERE sn = 5) [s\_pn]) AS parts\_1,

((spj [pn, jn]) GROUP [pn, COUNT(jn)] AS count\_jn) AS parts\_2:

((parts\_1 TIMES parts\_2) WHERE pn = s\_pn) [pn, count\_jn]

// parts\_1 – выбираем номера деталей, поставляемых поставщиком sn = 5

// parts\_2 – выбираем все уникальные сочетания деталей pn и проектов jn

// parts\_2\_cnt – подсчитываем количество проектов для каждой детали

// выбираем детали parts\_1 и количество проектов для каждой из них

WITH parts\_1 AS (SELECT DISTINCT pn AS s\_pn FROM spj WHERE sn = 5),

parts\_2 AS (SELECT DISTINCT pn, jn FROM spj),

parts\_2\_cnt AS (SELECT pn, COUNT(jn) AS count\_jn FROM parts\_2 GROUP BY pn)

SELECT pn, count\_jn FROM parts\_2\_cnt, parts\_1

WHERE parts\_2\_cnt.pn = parts\_1.s\_pn;