

# Průvodní dokument k SQL projektu

## 1) vytvoření základní tabulky s následujícími údaji:

- Datum
- název státu
- počet provedených testů v daném dni
- počet potvrzených případů
- rozlišení, zda se jednalo o víkend či pracovní den (zde byla použita přesnější varianta pro známý časový úsek, uvedena v komentáři je i varianta méně přesná, nicméně jednoduše aplikovatelná na libovolné časové období)

```
CREATE TABLE semitable_ok_base AS
SELECT
  DISTINCT
    cbd.date,
    cbd.country,
    tests.tests_performed,
    cbd.confirmed,
    CASE WHEN WEEKDAY(cbd.date) IN (5, 6) THEN 1 ELSE 0 END AS weekend_flag,
    CASE
      WHEN cbd.date BETWEEN '2019-12-21' AND '2020-03-20' THEN 3
      WHEN cbd.date BETWEEN '2020-03-21' AND '2020-06-20' THEN 0
      WHEN cbd.date BETWEEN '2020-06-21' AND '2020-09-22' THEN 1
      WHEN cbd.date BETWEEN '2020-09-23' AND '2020-12-20' THEN 2
      WHEN cbd.date BETWEEN '2020-12-21' AND '2021-03-20' THEN 3
      WHEN cbd.date BETWEEN '2021-03-21' AND '2021-06-20' THEN 0
      WHEN cbd.date BETWEEN '2021-06-21' AND '2021-09-22' THEN 1
      WHEN cbd.date BETWEEN '2021-09-23' AND '2021-12-20' THEN 2
      WHEN cbd.date BETWEEN '2021-12-21' AND '2022-03-20' THEN 3
      ELSE 'chyba'
    END AS season_of_year, -- kde 0 = jaro, 3 = zima
    -- varianta kratší a univerzálnější, ale méně přesná by mohla být:
    -- CASE
    --   when month(cbd.date) in (3, 4, 5) then 0
    --   when month(cbd.date) in (6, 7, 8) then 1
    --   when month(cbd.date) in (9, 10, 11) then 2
    --   when month(cbd.date) in (12, 1, 2) then 3
    -- end) as season
    led.life_expectancy_diff AS 'rozdil_dozeni_1965/2015'
FROM covid19_basic_differences AS cbd
LEFT JOIN covid19_tests AS tests
  ON cbd.country = tests.country
  AND cbd.date = tests.date
LEFT JOIN life_expectancy_diff AS led
  ON cbd.country = led.country
ORDER BY date
;
```

2) V dalším kroku byla vytvořena tato Views, která nám z tabulek „countries“ a „economies“ vracela námi požadovaná data:

- hustota zalidnění
- celková populace
- medián věku obyvatel
- HDP na obyvatele
- dětskou úmrtnost do 5 let věku
- GINI koeficient

Zbytek dat pro nás nebyl relevantní.

```
CREATE OR REPLACE VIEW v_ok_countries AS
SELECT
    c.country,
    c.population_density AS hustota_zalidneni,
    c.population AS populace,
    c.median_age_2018 AS medián_věku_2018
FROM countries AS c
;

CREATE OR REPLACE VIEW v_ok_economies AS
SELECT
    DISTINCT
    eco.country,
    eco.GDP/eco.population AS HDP_na_obyvatele,
    eco.mortality_under5 AS detska_umrtnost,
    eco.gini AS Gini_coef
FROM economies AS eco
WHERE YEAR = 2018
;
```

3.1) Pro vytvoření reportu o počasí v jednotlivých státech bylo potřeba vytvořit pomocné view, které nám vracelo státy k hlavním městům, kvůli JOINům v dalším kroku (abychom měli přes co JOINovat). Protože se v tabulkách „countries“ a „weather“ jména některých měst mírně lišily (dle jazykové mutace), bylo potřeba je upravit -viz SQL dotaz.

```
CREATE OR REPLACE VIEW v_stat_mesto AS
SELECT
    DISTINCT
    country,
    CASE
    WHEN capital_city = 'Wien' THEN 'Vienna'
    WHEN capital_city = 'Bruxelles [Brussel]' THEN 'Brussels'
    WHEN capital_city = 'Praha' THEN 'Prague'
    WHEN capital_city = 'Helsinki [Helsingfors]' THEN 'Helsinki'
    WHEN capital_city = 'Athenai' THEN 'Athens'
    WHEN capital_city = 'Roma' THEN 'Rome'
    WHEN capital_city = 'Luxembourg [Luxemburg/L]' THEN 'Luxembourg'
    WHEN capital_city = 'Warszawa' THEN 'Warsaw'
    WHEN capital_city = 'Bucuresti' THEN 'Bucharest'
    WHEN capital_city = 'Kyiv' THEN 'Kiev'
    ELSE capital_city END AS capital_city
FROM countries c
;
```

3.2) V dalším kroku jsme si z tabulky „weather“, která obsahuje data o počasí v jednotlivých hlavních městech, pomocí dotazů níže vyextrahovali jednotlivá views, která nám pro daná města počítala průměrnou teplotu za daný den (v\_weather\_AVGTEMP), počet hodin, kdy srážky nebyly nulové (v\_weather\_srazky) a maximální sílu větru v nárazech za daný den (v\_weather\_narazy). Následně byla tato views spojena v jeden report.

```

CREATE OR REPLACE view v_weather_narazy AS
SELECT
    DATE(`date`) AS datum,
    city,
    MAX(gust) AS vitr_narazy
FROM weather w
WHERE TIME(`time`) BETWEEN '00:00' AND '21:00' AND city IS NOT NULL
GROUP BY DATE(`date`), city;

CREATE OR REPLACE view v_weather_AVGTEMP AS
SELECT
    DATE(`date`) AS datum,
    city,
    AVG(REGEXP_SUBSTR(temp, '[0-9]+')) AS průměrná_denní_teplota
FROM weather w
WHERE TIME(`time`) BETWEEN '06:00' AND '18:00' AND city IS NOT NULL
GROUP BY DATE(`date`), city
;

CREATE OR REPLACE view v_weather_srazky AS
SELECT
    DATE(`date`) AS datum,
    city,
    SUM(CASE WHEN rain != "0.0 mm" THEN 3 ELSE 0 END) AS hodin_deste
    -- SUM(destivy_usek) AS pocet_hodin_deste_denne
FROM weather w
WHERE TIME(`time`) BETWEEN '00:00' AND '21:00' AND city IS NOT NULL
GROUP BY DATE(`date`), city
;

```

```

CREATE OR REPLACE VIEW v_ok_weather_report AS
SELECT
    sm.country,
    wn.datum,
    wn.city AS město,
    wn.vitr_narazy,
    wa.průměrná_denní_teplota,
    ws.hodin_deste
FROM v_weather_narazy wn
LEFT JOIN v_weather_avgtemp wa
    ON wn.datum = wa.datum
    AND wn.city = wa.city
LEFT JOIN v_weather_srazky ws
    ON wn.datum = ws.datum
    AND wn.city = ws.city
LEFT JOIN v_stat_mesto sm
    ON wn.city = sm.capital_city
;

```

4) Poslední sadou dat, která zadavatele zajímala, bylo procentuální zastoupení jednotlivých náboženství pro každou zemi. Bylo vytvořeno pomocné view, které nám do tabulky religions připojilo sloupec s celkovou populací daných států, na základě kterého jsme počítali procentuální zastoupení každého náboženství (níže uveden příklad pro křesťanství, identicky postupováno s dalšími náboženstvími). Nakonec byla všechna náboženství spojena do jedné tabulky s názvem „nabozenska\_statistika“.

```
-- ***** RELIGION STUFF ***** --

CREATE OR REPLACE VIEW v_religion_stat AS
SELECT
    r.country,
    c.population AS celkova_populace,
    r.religion AS nabozenstvi,
    r.population AS pocet_vericich
FROM religions r
LEFT JOIN countries c
    ON r.country = c.country
WHERE r.`year` = 2020;

-- podíl Křesťanství
CREATE OR REPLACE VIEW v_crist_share AS
SELECT
    country,
    (pocet_vericich/celkova_populace)*100 AS podíl_křesťanství_v_prc
FROM v_religion_stat vrs
WHERE nabozenstvi = 'Christianity'
;
```

```
CREATE TABLE ok_nabozenska_statistika AS
SELECT
    vcs.country,
    vcs.podíl_křesťanství_v_prc,
    vis.podíl_Islámu_v_prc,
    vbs.podíl_Buddhismu_v_prc,
    vhs.podíl_Hinduismu_v_prc,
    vjs.podíl_Judaismu_v_prc,
    vfs.podíl_Folk_rel_v_prc
FROM v_crist_share vcs
LEFT JOIN v_islam_share vis
    ON vcs.country = vis.country
LEFT JOIN v_buddhism_share vbs
    ON vcs.country = vbs.country
LEFT JOIN v_hinduism_share vhs
    ON vcs.country = vhs.country
LEFT JOIN v_judaism_share vjs
    ON vcs.country = vjs.country
LEFT JOIN v_folk_share vfs
    ON vcs.country = vfs.country
;
```

5) V posledním kroku byly všechny zmiňované podprodukty spojeny do jedné přehledové tabulky s názvem **FINAL\_OVERVIEW\_KNOPP**.

```
-- *** BRINGING ALL TOGETHER *** --  
  
CREATE TABLE FINAL_OVERVIEW_KNOPP AS  
SELECT  
  DISTINCT  
    sob.`date`,  
    sob.country,  
    sob.tests_performed,  
    sob.confirmed,  
    sob.weekend_flag,  
    sob.season_of_year,  
    voc.hustota_zalidneni,  
    voc.populace,  
    voc.medián_věku_2018,  
    voe.HDP_na_obyvatele,  
    voe.detska_umrtnost,  
    voe.Gini_coef,  
    ons.podíl_křesťanství_v_prc,  
    ons.podíl_Islámu_v_prc,  
    ons.podíl_Buddhismu_v_prc,  
    ons.podíl_Hinduismu_v_prc,  
    ons.podíl_Judaismu_v_prc ,  
    vowr.vitr_narazy,  
    vowr.průměrná_denní_teploata,  
    vowr.hodin_deste  
FROM semitable_ok_base sob  
LEFT JOIN v_ok_countries voc  
  ON sob.country = voc.country  
LEFT JOIN v_ok_economies voe  
  ON sob.country = voe.country  
LEFT JOIN ok_nabozenska_statistika ons  
  ON sob.country = ons.country  
LEFT JOIN v_ok_weather_report vowr  
  ON sob.country = vowr.country  
  AND sob.`date` = vowr.datum  
;
```