Τμήμα 1 | Εξερεύνηση του World Bank Dataset μέσω του Colaboratory (40 μονάδες)

Υποδείξεις - διαβάστε τις πολύ προσεκτικά! - :

- Σιγουρευτείτε ότι διαβάσατε καλά τις οδηγίες σε κάθε κελί και κατανοήσατε τι υλοποιεί πριν το εκτελέσετε.
- Να θυμάστε ότι έχετε τη δυνατότητα να μεταφορτώνετε το αρχικό "σημειωματάριο" όποτε το χρειάζεστε.
- Μπορείτε να δημιουργείτε νέα κελιά για να τα χρησιμοποιείτε σε ελέγχους, εκσφαλμάτωση, εξερεύνηση κλπ. Μάλιστα προτείνουμε να το κάνετε! **Βεβαιωθείτε εντούτοις ότι η τελική απάντηση σε κάθε ερώτηση βρίσκεται στο *δικό της κελί* και *προσδιορίζεται ρητά***.
- Το Colaboratory δεν σας ειδοποιεί για τα bytes που θα καταναλώσει η εκτέλεση των SQL ερωτημάτων σας. **Σιγουρευτείτε ότι ελέγχετε την κατανάλωση μέσω της διεπαφής (UI) του BigQuery πριν εκτελέσετε τα ερωτήματά σας στο Colaboratory!**
- Ακολουθείστε τις οδηγίες υποβολής.

Μέλη της Ομάδας Εργασίας:

Παραθέστε τα ονοματεπώνυμα και τους ΑΜ των μελών της ομάδας στην ακόλουθη λίστα:

Καραθανάσης Δημήτρης, 3547

Συλλιγάρδος Εμμανουήλ, 3849

Ρυθμίσεις για το BigQuery και τις σχετικές εξαρτήσεις

Εκτελέστε τα δύο ακόλουθα κελιά (shift + enter) προκειμένου να πιστοποιήσετε την εργασία σας και να φορτώσετε τις απαιτούμενες βιβλιοθήκες.

Προσέξτε ότι θα χρειαστεί να συμπληρώσετε τη μεταβλητή `project_id` στο πρώτο κελί με το Google Cloud Project ID που έχετε δημιουργήσει για τις ανάγκες της εργασίας σας. Για να δείτε το project ID μεταβείτε στη σελίδα https://console.cloud.google.com/cloud-resource-manager.

```
# Εκτελέστε αυτό το κελί προκειμένου να πιστοποιήσετε την εργασία σας στο BigQuery.
from google.colab import auth
auth.authenticate_user()
project_id = 'ancient-tractor-226920'
```

```
# Βιβλιοθήκες που θα χρειαστείτε
import pandas as pd
import altair as alt
```

Χρήση του BigQuery στο Collab

Τα σημειωματάρια στο Jupyter (στα οποία βασίζονται τα σημειωματάρια του Collab) χρησιμοποιούν τη ιδέα της "μαγείας". Εάν γράψετε την ακόλουθη γραμμή στην κορυφή ενός κελιού με 'Κώδικα' :

```
%%bigquery --project $project_id variable # this is the key line
SELECT ....
FROM ...
```

το "%%" μετατρέπει το κελί σε κελί SQL. Ο πίνακας που παράγεται από το ερώτημα αποθηκεύεται στη μεταβλητή `variable`. Στη συνέχεια εάν γράψετε σε δεύτερο κελί:

```
alt.Chart(variable).mark_line().encode(
...
)
```

μπορείτε να χρησιμοποιήσετε τη μεταβλητή ώστε να δημιουργήσετε ένα γράφημα!

Ενότητα 1 | Σχεδίαση του Σχήματος!

Ο οργανισμός World Bank συλλέγει και συγκεντρώνει δεδομένα από πολλές δημόσιες πηγές ανά τον κόσμο και τα δημοσιεύει για ηλεκτρονική πρόσβαση. Το BigQuery μας παραχωρεί τα δεδομένα αυτά για να τα επεξεργαστούμε, ενώ περιέχει ένα μεγάλο αριθμό από μετρικές (δείκτες) σχετικές με δραστηριότητες και συμπεράσματα για διάφορα έθνη.

Για την εργασία αυτή θα χρησιμοποιήσουμε το δημόσιο σύνολο δεδομένων `world_bank_health_population`.

Ερώτηση 1: Περιγράψτε το σύνολο δεδομένων World Bank (1 μονάδα)

Εάν έπρεπε να περιγράψετε το τρόπο με τον οποίο έχουν οργανωθεί τα δεδομένα στα σύνολα του World Bank (οποιοδήποτε από τα τέσσερα καθώς έχουν ίδια δομή), τι θα λέγατε; **Σημείωση:** Τα ερωτήματα που ακολουθούν αναφέρονται διεξοδικότερα στη δομή του συνόλου δεδομένων, επομένως εδώ ζητούμε μια επιγραμματική αναφορά. Θέλουμε τις εντυπώσεις σας - τι παρατηρήσατε;

Στα country tables εμφανιζονται πληροφοριες για την καθε χωρα καθως και οι ορισμοι για καθε δεικτη(health population, education κλπ). Τα series tables(summary, times) αναφερονται στην γενικη επισκοπηση των δεικτων καθως και στο χρονικο διαστημα που αυτοι αναφερονται.

Γνωριμία με το OKV, το Αντι-Σχήμα

Τα αρχικά **`OKV`** σημαίνουν `Object-Key-Value` [1]. Πρόκειται για ένα τρόπο αποθήκευσης δεδομένων ακριβώς αντίθετο από αυτόν που βασίζεται σε σχήματα: έχετε την ελευθερία να ορίσετε οποιοδήποτε γνώρισμα επιθυμείτε σε οποιοδήποτε αντικείμενο. Σκεφτείτε ότι φτιάχνετε ένα γιγαντιαίο πίνακα κατακερματισμού (10 δισ. γραμμές είναι λίγες σε αυτόν [2]) για κάθε μεταβλητή

γιγαντιαίο πίνακα κατακερματισμού (τα στο: γραμμές είναι λίγες σε αυτόν [2]) για κάθε μεταβλητή αντικειμένου στο σύστημά σας.

Ακολουθεί ένας τρόπος με τον οποίο θα μπορούσε να αναπαρασταθεί ένας τέτοιος πίνακας:

object	key	value
102	"name"	"John Watson"
103	"name"	"Sherlock Holmes"
102	"address"	"221B Baker Street, London, UK"
107	"name"	"Oprah Winfrey"
103	"address"	"221B Baker Street, London, UK"
102	"canes"	26
103	"cases_solved"	60

Όπως παρατηρείτε, τα τρία αντικείμενα του πίνακα έχουν διαφορετικές "μορφές" (όρος που χρησιμοποιείται αντί για το "σχήμα" στις περιπτώσεις που δεν ακολουθείται ένα τυπικό σχήμα).

Εάν θέλετε να μάθετε για κάποιο αντικείμενο θα πρέπει να κάνετε μια επερώτηση όπως παρακάτω :

```
SELECT key, value
FROM table
WHERE object = 102
```

Στη συνέχεια η συγχώνευση όλων των απαντήσεων θα σας δώσει τη συνολική πληροφορία για το αντικείμενο!

Παρατηρήσεις

1. Άλλες εκδοχές της ιδέας που συζητούμε (χρήση τριών λέξεων για την αποθήκευση δεδομένων) περιλαμβάνουν τις: ID-Key-Value, Object-Property-Value, Entity-Attribute-Value, Entity-Property-Value, για τις οποίες υπάρχουν αντίστοιχα ακρωνύμια IKV, OPV κλπ.
2. Ο λόγος ύπαρξης μεγάλου αριθμού γραμμών σε αποθήκες OKV, είναι ότι περιλαμβάνουν όλα τα κελιά ενός κανονικού πίνακα (που ακολουθεί κάποιο σχήμα).

Επιπλέον μελέτη (Ενδεικτική)

- Άρθρο στη Wikipedia για τη συγκεκριμένη δομή αποθήκευσης

Ερώτηση 2: Ασχολούμαστε με τα OKVs (6 μονάδες)

Συγκρίνετε τις αποθήκες OKV με τους "κλασσικούς" σχεσιακούς πίνακες. Ποια είναι τα πλεονεκτήματά τους; Ποιες οι δυσκολίες τους;

(Απαντήστε με 200 το πολύ λέξεις - προτείνουμε λίστα με κουκκίδες!)

Υποδείξεις

- Το ακρωνύμιο **CRUD** ορίζει μια χρήσιμη αναφορά ελέγχου, με τα αρχικά του να αντιστοιχούν στις βασικές ενέργειες που εφαρμόζονται στα δεδομένα: **Create**, **Read**, **Update**, **Delete**.

Μπορείτε να δημιουργείτε/διαβάζετε /ενημερώνετε/διαγράφετε τιμές σε μια ΒΔ, ή στο σχήμα της (πχ. προσθήκη/διαγραφή ενός γνωρίσματος, αλλαγή του τύπου του κλπ).

- Όταν σκέφτεστε για πλεονεκτήματα και μειονεκτήματα στο λογισμικό, ορισμένα κοινώς επιθυμητά χαρακτηριστικά είναι η επίδοση (χρόνος εκτέλεσης των επερωτήσεων), το αποτύπωμα στη μνήμη (όσο λιγότερη μνήμη χρησιμοποιείται, τόσο καλύτερα), η διατήρηση (εάν μπορεί η ΒΔ να προσαρμοστεί εύκολα στις απαιτήσεις των εφαρμογών) και η πολυπλοκότητα του κώδικα (εάν η σχεδίαση της ΒΔ ενθαρρύνει τη δημιουργία μεγάλων, δυσμεταχείριστων επερωτήσεων κάτι που μπορεί να οδηγήσει σε προγραμματιστικά λάθη λόγω πολυπλοκότητας). Η σύγκρισή σας μπορεί να αναφέρεται σε αυτά τα χαρακτηριστικά για καθεμιά από τις παραπάνω βασικές ενέργειες (CRUD) στις δυο περιπτώσεις οργάνωσης.
- Για τις επιδόσεις στις ΒΔ, χάριν της ερώτησης αυτής, μπορείτε να σκέφτεστε σε τρία επίπεδα:
    - Εντοπισμός: Έχετε μια τιμή κλειδιού ενός πίνακα και ψάχνετε μια γραμμή του (ή κάποιο υποσύνολό της). Θεωρήστε ότι έχει πλοκή O(1).
    - Σάρωση: Όταν πρέπει να εντοπίσετε γραμμές βάσει κριτηρίων (πχ, άνθρωποι ψηλότεροι από 1,80). Το ύψος δεν είναι κλειδί, κι έτσι πρέπει να ψάξετε όλες τις γραμμές του πίνακα. Θεωρήστε ότι έχει πλοκή O(N). Ανάλογα ενεργείτε όταν πρέπει να δώσετε τιμές σε ένα γνώρισμα σε πολλές γραμμές μαζί.
    - Σύζευξη: Όταν γίνεται σύζευξη πινάκων, δημιουργείται ένα καρτεσιανό γινόμενο συνόλων.Εάν ο ένας πίνακας έχει N γραμμές και ανάλογα συμβαίνει και με τον δεύτερο, η σύζευξη θεωρείται ότι έχει πλοκή O(N^2).

Τα OKV μοντέλα χρησιμοποιούνται για να αποθηκευθούν, με memory-efficient τρόπο, αντικείμενα που ο αριθμός των χαρακτηριστικών τους είναι μεγάλος και διαφέρει από object σε object. Εάν τα αντικείμενα που θέλουμε να αποθηκεύουμε δεν πληρούν αυτό το χαρακτηριστικό, τότε η χρήση κλασικών σχεσικών μοντέλων είναι προτιμότερη.

Πλεονεκτήματα του OKV μοντελου:

- Space efficiency σε object που αντιστοιχούν σε sparse matric μαθηματικά μοντέλα.
- Ο εντοπισμός ενός object και το χαρακτηριστικών του, έχοντας το κλειδί του object, γίνεται σε O(1) χρόνο έναντι του αντίστοιχου O(n) που θα χρειαζόταν η ίδια αναζήτηση σε ένα αντίστοιχο σχεσιακό μοντέλο.

Μειονεκτηματα του OKV μοντελου:

- Η αναζήτηση οντοτήτων με βάση τα χαρακτηριστικά τους γινέται σε O(n), όπως και στα σχεσιακά μοντέλα, όμως το n εδώ είναι πολύ μεγαλύτερο αφού τα OKV μοντέλα είναι μακρόστενα, δηλαδή έχουν πάρα πολλά rows.
- Το καρτεσιανό γινόμενο 2 OKV πινάκων είναι πολύ μεγαλύτερο από το καρτεσιανό 2 σχεσιακών πινάκων.

- Η συντήρηση μίας OKV βάσης δεδομένων είναι δυσκολότερη, διότι χρειάζεται βοηθητικούς πίνακες. Οι βοηθητικοί πίνακες κρατούν metadata πληροφορία που χρειάζονται οι OKV πίνακες για να λειτουργήσουν. Κάθε OKV πίνακας χρειάζεται περίπου 3 ή περισσότερους κανονικούς σχεσιακούς πίνακες με metadata πληροφορία

κανονικούς σχεσιακούς πίνακες με metadata πληροφορία.

- Σε αντίθεση με τα σχεσικά μοντέλα, τα OKV μοντέλα είναι ανομοιογενή. Κάθε σειρά περιγράφει οποιουδήποτε είδους πληροφορία για την αντίστοιχη οντότητα. Οι τιμές δεν έχουν στανταρ τύπο, αλλά ο τύπος τους εξαρτάται απόλυτα απο το χαρακτηριστικό που περιγράφουν.

Γενικά χρησιμοποιούμε OKV μοντέλα μόνο όπου χρειάζεται. Δηλαδή οπουδήποτε η πληροφορία που θέλουμε να αποθηκεύσουμε δεν χαρακτηρίζεται από πολλά και διάσπαρτα attributes, τότε χρησιμοποιούμε κανονικά σχεσιακά μοντέλα.

Κάτι ακόμα - Ονόματα γνωρισμάτων

Όπως έχετε μάθει, ο πλεονασμός δεδομένων στους πίνακες είναι ανεπιθύμητος, καθώς, εάν θέλετε να αλλάξετε μια τιμή, πρέπει να ενημερώσετε όλα τα σημεία στα οποία εμφανίζεται (πολύ ακριβή ενέργεια - θυμηθείτε ότι στο μοντέλο OKV ένας πίνακας έχει πολύ περισσότερες γραμμές απ' ότι στο σχεσιακό). Κάτι τέτοιο είναι επίσης γνωστό ως **ανωμαλία ενημέρωσης**.

Πώς χειρίζεστε αυτό το θέμα; Είναι απλό: με τη χρήση πίνακα γνωρισμάτων:

```
# Schema (με βάση κάποια σύνταξη):
Property(id, name)
Data(id, key, value)
```

Έτσι θα αντικαθιστούσαμε τον παραπάνω πίνακα με τον:

Πίνακας γνωρισμάτων:

id	name
1	"name"
2	"address"
3	"canes"
4	"cases_solved"

Πίνακας δεδομένων:

id	pid	value
102	1	"John Watson"
103	1	"Sherlock Holmes"
102	2	"221B Baker Street, London, UK"
107	1	"Oprah Winfrey"
103	2	"221B Baker Street, London, UK"
102	3	26
103	4	60

Ερώτηση 3: Επανερχόμαστε ... (2 μονάδες)

Επαναλάβετε τη σύγκρισή σας για τον πίνακα γνωρισμάτων - σε τι διευκολύνει η αλλαγή που προτάθηκε παραπάνω και τι είναι ακόμη δύσκολο να γίνει;

Παρακαλώ σχολιάστε μόνο τις διαφορές - μην επαναλάβετε την ανάλυση.

Κάνοντας χρήση πίνακα γνωρισμάτων βελτιώνουμε την αναζήτηση και την ενημέρωση στους πίνακες δεδομένων. Ο OKV πίνακας δεδομένων έχει πιο καθαρή μορφή με τα γνωρίσματα του να έχουν όλα το δικό τους ID και να μην είναι απλά μικρά περιγραφικά strings. Η αναζήτηση γίνεται πιο γρήγορα αφού ψάχνουμε κάθε φορά το χαρακτηριστικό μίας οντότητας έχοντας 1 μοναδικό κλειδί που αποτελείτε από id και pid. Η ενημέρωση γίνεται πιο γρήγορα για τον ίδιο λόγο, αφού κάθε ενημέρωση αποτελείται από μία αναζήτηση και την αλλαγή της τιμής που θέλουμε. Από την άλλη για την δημιουργία και την διαγραφή ενούς row δεν αλλάζει κάτι όσον αφορά στο χρόνο που χρειάζονται. Τα μειονεκτήματα είναι, ότι ο νέος πίνακας γνωρισμάτων καταναλώνει περισσότερο χώρο στη μνήμη και η εισαγωγή ενός νέου χαρακτηριστικού ή η ολοκληρωτική διαγραφή του απαιτεί την ενημέρωση 2 πινάκων έναντι του ενός που απαιτούσαι προηγουμένως.

Ένα ακόμη πράγμα - Οι τύποι!

Στην SQL, κάθε στήλη πρέπει να έχει έναν τύπο [1]. Έτσι όταν άρχισαν να αναμιγνύονται σε μια στήλη *string* με *int* τιμές, αυτό ήταν απλοποίηση.

Υπάρχουν πολλές σχεδιαστικές επιλογές για την επίλυση αυτού του ζητήματος - δείτε την επόμενη ερώτηση όπου συζητούνται ορισμένες από τις επιλογές αυτές.

Σημείωση

Υπάρχουν ΒΔ που δεν έχουν το χαρακτηριστικό που παρουσιάστηκε πιο πάνω: να μπορεί να καταχωρούνται τιμές οποιουδήποτε τύπου στην τρίτη στήλη. Εάν απορείτε γιατί όλοι επιθυμούν αυτό το χαρακτηριστικό, σκεφτείτε τη διαφορά μεταξύ γλωσσών προγραμματισμού με στατικά και δυναμικά ορισμένους τύπους (πχ Java vs Python). Ανάλογα αντισταθμίζεται η επιλογή μιας ΒΔ που το σχήμα της έχει γνωρίσματα καθορισμένων εξαρχής τύπων απ' ότι άλλης ΒΔ που το σχήμα της έχει γνωρίσματα ακαθόριστων τύπων.

Ερώτηση 4: "Διάλογος μεταξύ φίλων" (6 μονάδες)

(ΥΓ - Το ρωτούν συχνά σε συνεντεύξεις μηχανικών λογισμικού!).

Ένας καλός σας φίλος προσπαθεί να υλοποιήσει μια αποθήκη OKV σε SQL και συναντά το εμπόδιο που περιγράψαμε προηγουμένως. Ας θεωρήσουμε απλουστευτικά ότι ενδιαφέρεται να αποθηκεύει μόνο *string* και *int* τιμές (μπορεί να επεκταθεί και για άλλους τύπους).

Προτείνεται η ακόλουθη λύση (αν και δε φαίνεται, υποθέστε ότι ο πίνακας γνωρισμάτων - για τα pid - υπάρχει επίσης):

id	pid	string_value	int_value
102	1	"Sherlock Holmes"	null
103	1	"John Watson"	null
102	3	null	60

Επεξηγηματικά: εάν η τιμή έχει τύπο *string*, συμπληρώνεται κατάλληλα η στήλη `string_value` και στη στήλη `int_value` μπαίνει null και ανάλογα για τιμές τύπου *int*.

α) Τι λάθος εντοπίζετε στον παραπάνω πίνακα; (2 μονάδες)

Εάν έπρεπε να κρίνετε την πρόταση αυτή, τι θα λέγατε στον φίλο σας; Ποια η δυσκολία και τα ανεπιθύμητα χαρακτηριστικά της;

Το λάθος που εντοπίζουμε είναι ότι ο φίλος κάνει λάθος χρήση του OKV μοντέλου εξ ορισμού. Τα OKV μοντέλα χρησιμοποιούνται για να αποφύγουμε να έχουμε ένα πίνακα με πολλά κενά (null) κελιά. Ο φίλος μας εδώ, χρησιμοποιεί OKV πίνακα και έχει και 1 κενό κελί σε κάθε σειρά με αποτέλεσμα να σπαταλάει άδικα πάρα πολύ μνήμη.

β) Η αντιπρότασή σας (2 μονάδες)

Προτείνετε μια άλλη σχεδίαση στην οποία αντιμετωπίζονται προβλήματα που περιγράψατε στην προηγούμενη απάντησή σας (υπόδειξη: ίσως χρειαστεί περισσότερους από έναν πίνακες).

Προτείνουμε στο φίλο μας να χρησημοποιήσει 2 OKV πίνακες αντί για έναν. Ένα πίνακα στον οποίο θα αποθηκεύονται οι οντότητες με όλα τα χαρακτηριστικά τους που έχουν τιμές τύπου int και έναν άλλο πίνακα που αποθηκεύονται οι οντότητες με τα χαρακτηριστικά τους που έχουν τιμές τύπου string. Έτσι δεν θα υπάρχει κανένα κενό κελί στους 2 πίνακες.

γ) Αντι-κριτική! (2 μονάδες)

Ο φίλος σας εξετάζει τη δική σας πρόταση και τη σχολιάζει.Σε ποιες δυσκολίες και ανεπιθύμητα χαρακτηριστικά της θα αναφερθεί;

Κάθε φορά που θέλουμε να διαγράψουμε μία οντότητα πρέπει να ενημερώνουμε και τους 2 πίνακες. Ακόμη αν θέλουμε να βρούμε όλα τα χαρακτηριστικά μιας οντότητας πρέπει να διατρέξουμε σε 2 πίνακες αντί για έναν. Τέλος παρόλο που η πρόταση μας δεν έχει κενά κελιά, αν περίπου κάθε οντότητα που εισάγουμε στη βάση έχει και ενα χαρακτηριστικό τύπου int και ένα χαρακτηριστικό τύπου string, τότε καταναλώνουμε περισσότερο χώρο στη μνημη από την αρχική πρόταση του φίλου μας.

Εφαρμόστε αυτά που μάθατε από τα προηγούμενα

Τα δεδομένα της world bank έχουν τη δομή OKV... με μια μικρή διαφορά. Οι πίνακες που περιέχουν τα δεδομένα έχουν λίγο πολύ την ακόλουθη μορφή:

```
object | key | year | value
```

όπου `object` = κωδικός χώρας & `key` = κωδικός δείκτη (τι μετρήθηκε).

Υπάρχουν λίγα ακόμη γνωρίσματα (περιγραφές για object και key), αλλά συνολικά η δομή είναι OKV

Υπάρχουν κι για άλλη γνώρισματα (περιγραφές για object και key), αλλά συνολικά η δομή είναι OKV.

Με τη γνώση που αποκτήσατε από τα παραπάνω, δείτε ξανά το σχήμα και εντοπίστε και άλλες ιδιότητες της δομής αποθήκευσης key-value που προσδιορίσαμε.

Ερώτηση 5: Κατανόηση του σχήματος (3 μονάδες)

Καθένα από τα παρακάτω παίρνει 1 μονάδα.

α) Ποιος πίνακας, μεταξύ των τεσσάρων που περιλαμβάνει το σύνολο δεδομένων της world bank, έχει το ρόλο του πίνακα γνωρισμάτων;

Σε κάθε ένα από τα 4 data sets ο πίνακας που έχει τον ρόλο του πίνακα γνωρισμάτων είναι ο πίνακας "series_summary".

β) Ποιος πίνακας περιέχει παραπάνω πληροφορία για τα "αντικείμενα" (σε συμφωνία με τη δομή OKV);

Σε κάθε ένα από τα 4 data sets ο πίνακας που περιέχει παραπάνω πληροφορία για τα αντικείμενα είναι ο πίνακας "country_summary".

γ) Ποιο είναι το κλειδί ("key"), σε συμφωνία με τη δομή OKV, του πίνακα `health_nutrition_population`;

Το κλειδί στον πίνακα "health_nutrition_population" αποτελείται από τα υποκλειδιά γνωρίσματα "country_code" και "indicator_code".

Ερώτηση 6: Θεωρία σχεδίασης (12 μονάδες)

Δώστε το δικό σας σχήμα για τα δεδομένα της world bank! Στόχος είναι να φανταστείτε πώς θα μπορείτε να απαντάτε σε ερωτήματα όπως τα παρακάτω:

- Πώς εξελίσσεται στο χρόνο η ανάλυση του πληθυσμού στις ΗΠΑ σε ανδρικό και γυναικείο ανά δεκαετία (0-9, 10-19, 20-29, κλπ);
- Παρατηρείται δημογραφική γήρανση ή ανανέωση στην Ελλάδα;
- Πώς διαφοροποιείται η ανάλυση του πληθυσμού των ΗΠΑ και της Ελλάδας (ή άλλων χωρών);
- Ποιο είναι το προσδόκιμο ζωής σε σχέση με τις ιατρικές δαπάνες για όλες τις χώρες του κόσμου;
- Σε ποιες περιοχές εξαπλώνεται ο HIV; Φτιάξτε εικόνα της κατανομής των ασθενών με AIDS για περιοχές με υψηλά ποσοστά της ασθένειας.

Επιπλέον απαιτήσεις

- Ανεξάρτητα από το σχήμα που θα προτείνετε, θα πρέπει να είναι σαφής ο τρόπος εισαγωγής δεδομένων σε διάφορα επίπεδα: πλειάδες (γραμμές πινάκων), γνωρίσματα (στήλες πινάκων) και πίνακες.
    - πχ, εάν αρχικά δεν αποθηκεύατε το κατά κεφαλήν ΑΕΠ, πώς θα το προσθέσετε στον κατάλληλο πίνακα;

Υποδείξεις:

Η πραγματικότητα ξεπερνάει κάθε φαντασία, επομένως κατά πάσα πιθανότητα και τις σχεδιαστικές σας επιλογές! Θα θέλαμε οι ΒΔ να ανταποκρίνονται στα γεγονότα από τα οποία δημιουργούνται τα δεδομένα τους. Θυμηθείτε τις ενέργειες CRUD (create, read, update, delete)! Τι μπορούμε να κάνουμε στις περιπτώσεις που:

- Μια στατιστική αποδειχθεί λανθασμένη και χρήζει αναθεώρησης;
- Χρειάζεται να προσθέσετε δεδομένα από όλες τις χώρες, τη στιγμή που δημιουργούνται, με το τέλος του 2018;
- Μια χώρα διχοτομείται μετά από επανάσταση;
- Μια χώρα αλλάζει το όνομά της;
- Χρειάζεται να αποθηκεύετε πολύ μικρά ποσοστά (πχ επικράτηση σπάνιων ασθενειών);
- Υπάρχουν στατιστικές που εφαρμόζονται μόνο σε ορισμένες χώρες (πχ, ποσοστό ανθρώπων που τηρούν το ραμαζάνι);
- Απροσδόκητα απαιτείται η αποθήκευση δεδομένων με μεγαλύτερο ρυθμό (ας πούμε εβομαδιαία ή μηνιαία, αντί ετησίων);

Είναι μάλλον απίθανο να συμπεριφέρεται καλά η σχεδίασή σας σε όλες τις παραπάνω περιπτώσεις (και πολλές ακόμη άλλες που θα σκεφτείτε), αλλά δεν είναι πρόβλημα! Δεν υπάρχει τέλεια σχεδίαση. Εντούτοις, θέλουμε να μας δείξετε ότι κατανοείτε πώς αντισταθμίζονται οι σχεδιαστικές επιλογές και τι σημαίνει αυτό για τις εφαρμογές που "χτίζετε" πάνω από τις ΒΔ σας.

α) Ποιες είναι οι οντότητες στο σχήμα σας; (2 μονάδες)

Country, Year , Population, Infected, Series.

Το Country και το Year αποτελούν μαζί μία υπεροντότητα.

Το Population είναι ασθενής οντότητα της υπεροντότητας Country_Year.

Το Infected είναι ασθενής οντότητα του Population.

β) Ποιες είναι οι μεταξύ τους σχέσεις; (Δε χρειάζεται να σχεδιάσετε ένα τέλειο διάγραμμα Οντοτήτων/Συσχετίσεων - αρκεί ένα βασικό που θα συνοδεύεται από λίστα με τις πληθικότητες για κάθε ζεύγος σχέσων '1 - 1', '1 - N' and 'N - N'). (2 μονάδες)

- Country ---(0,N)---<Λαμβάνει_Χώρα>---(0,N)--- Year
- Country_Year ---(0,N)---<Συνέβει>---(0,N)--- Series
- Country_Year ---(1,1)---<Έχει>---(1,1)--- Population
- Population ---(0,N)---<Είναι>---(1,1)--- Infected (%)

γ) Δώστε σχέδιο των πινάκων της ΒΔ σας (σαν αυτούς που εμφανίστηκαν προηγουμένως), και σημειώστε με ξεκάθαρο τρόπο ποια γνωρίσματα συνθέτουν το πρωτεύον κλειδί σε καθένα, καθώς επίσης και ποια γνωρίσματα είναι κλειδιά άλλων πινάκων (ξένα κλειδιά). (3 μονάδες)

- Country

country_id	country_name
GRC	Greece
AUS	Austria

- Year

year_value
1990
1991

- Λαμβάνει_Χώρα

country_id	year_value	life_expectancy	health_expenditure
PER	2000	44.7	24
ZMB	2000	70.31	90

- Population

country_id	year_value	gender	age_group	value
1990	Greece	male	0-14	7e+6
1991	Austria	female	15-64	8.773e+9

- Infected

country_id	year_value	infection_id	description	name	value(%)
1990	Greece	HIV	AIDS	Aids	0.10
1991	South Africa	YFV	Yellow Fever	Yellow Fever	1.2

- Series

series_id	series_name	description
SH.UHC.SRVS.CV.XD	UHC service coverage index	UHC service coverage index
SP.POP.GROW	Population growth (annual %)	Annual population growth rate

- Συνέβει

country_id	series_id	year_value	value
UZB	SL.UEM.TOTL.MA.ZS	2018	6.9

country_id	series_id	year_value	value
SL.UEM.TOTL.MA.ZS			

δ) Καταγράψτε τις (ελάχιστες) συναρτησιακές εξαρτήσεις κάθε πίνακα. (2 μονάδες)

- Country : country_id --> country_name
- Λαμβάνει_Χώρα : country_id , year_value --> life_expectancy , health_expenditure
- Population : country_id, year_value , gender , age_group --> value
- Infected : country_id , year_value, infection_id , description , name --> value
- Series : series_id --> series_name, description
- Συνέβει : country_id, series_id, year_value --> value

ε) Σχολιάστε τη σχεδίασή σας - Για ποιες περιπτώσεις είναι καλή/κακή; *Τι σταθμίσατε κατά τη διάρκεια λήψης των σχεδιαστικών σας επιλογών;* (3 μονάδες)

Η σχεδίαση μας προέκυψε καθώς προσπαθήσαμε να λύσουμε τα ερωτήματα που μας τέθηκαν στην αρχή της ερώτησης 6. Κάθε ένας από τους πίνακες είναι φτιαγμένος για να απαντά ένα ερώτημα τέτοιου τυπου. Στην σχεδίαση προστέθηκε τελικά και η οντότητα Series και η σχέση συνέβει έτσι ώστε το σχήμα να απαντά εντέλει και ερωτήματα όπως αυτά που κάνουμε εμείς στην world_bank_health_population. Η σχεδίαση μας, λοιπόν, μπορεί να απαντά ερωτήματα όπως εκείνα της ερώτησης 6, δηλαδή, ερωτήματα στοχευμένα και με την προσθήκη των 2 τελευταίων πινάκων μπορεί να απαντήση και σε ερωτήσης που απαντά και το σχήμα της world_bank.

Ενότητα 2 | Εξοικειωθείτε με την οπτικοποίηση

Στην ενότητα αυτή θα απαντήσετε σε ερωτήσεις όπως κάνατε στο 1ο μέρος της συνθετικής εργασίας (με χρήση SQL). Η διαφορά είναι ότι οι απαντήσεις σας θα οπτικοποιούνται. Μέρος της άσκησής σας είναι να σκεφτείτε ποιο είδος απεικόνισης (διάγραμμα, εικόνα κλπ) θα αποδώσει καλύτερα την απάντηση, καθώς επίσης και ποια δεδομένα ("μετρικές/δείκτες") θα χρησιμοποιήσετε για την απάντηση μια συγκεκριμένης ερώτησης.

Επικεντρωνόμαστε σε οπτικοποιήσεις καθώς πρόκειται για πρωτεύουσα μέθοδο κατανόησης και ερμηνείας της φύσης των δεδομένων. Ιδιαίτερα για τα "Μεγάλα Δεδομένα" που γίνεται λόγος στις μέρες μας, μια εικόνα αξίζει 1 εκατομμύριο γραμμές πίνακα :) .

Για μια γρήγορη ματιά στο τι μπορούμε να κάνουμε, δείτε το Gapminder. Αποτελεί εργαλείο για επαγγελματικού επιπέδου οπτικοποίηση μετρικών από δεδομένα, που επιπλέον είναι διαδραστικό!

Μπορείτε να αναζητήσετε αξιόλογες TED ομιλίες στις οποίες χρησιμοποιείται το Gapminder για την αναπαράσταση παγκόσμιων στατιστικών.

Εάν χρειάζεται να ελέγξετε "απαντήσεις" για κάποια σχεσιακά δεδομένα (δείτε: scatterplot), ψάξτε τα

στο Gapminder και βεβαιωθείτε ότι πήρατε μια απάντηση που μοιάζει σωστή. Όπως αναφέρθηκε, μέρος της εργασίας είναι η επιλογή των σωστών "δεικτών/μετρικών". Μπορείτε να "παίξετε" στο Gapminder με διαφορετικές περιπτώσεις πριν καταλήξετε στην επιλογή σας!

Γενικές οδηγίες

- Για καθεμιά από τις ερωτήσεις που ακολουθούν θα πρέπει να συμπληρώσετε τουλάχιστον δύο κελιά - ένα SQL στο οποίο εκτελείται το ερώτημά σας (και αποθηκεύει το αποτέλεσμα σε πλαίσιο δεδομένων), και ένα οπτικοποίησης όπου κατασκευάζετε το διάγραμμα αναπαράστασης του αποτελέσματος. Παρακαλώ έχετε κατά νου ότι ο χειρισμός των δεδομένων θα γίνει **αποκλειστικά** με SQL. Επίσης δεν πρέπει να χρησιμοποιήσετε τη βιβλιοθήκη `pandas` ή άλλη βιβλιοθήκη της python για να κάνετε μασάζ στα δεδομένα σας.
- Φτιάξτε τα διαγράμματά σας ευανάγνωστα - ετικέτες στους άξονες, ξεκάθαρα διακριτικά σημάδια, ευδιάκριτα σημεία/γραμμές/σχήματα, κλίμακες κλπ.
- Ψάξτε αρκετά τους δείκτες που θα χρησιμοποιήσετε. Εν τέλει μας ενδιαφέρει το διάγραμμα που θα προκύψει να έχει τη ζητούμενη πληροφορία - ακόμη κι αν την εμφανίζει με διαφορετικό τρόπο (πχ πληθυσμό ανά δεκαετία αντί για πληθυσμό ανά ηλικιακή ζώνη). Εντούτοις, κάποιοι δείκτες θα οδηγήσουν σε ευκολότερες λύσεις: για τούτο προτείνουμε να ξοδέψετε χρόνο για να εντοπίσετε αυτούς, που ο υπολογισμός τους θα γίνει με πιο άμεσο τρόπο.

Βιβλιοθήκες οπτικοποίησης

Τα σημειωματάρια του Colaboratory έχουν προεγκατεστημένη μια βιβλιοθήκη οπτικοποίησης που ονομάζεται **Altair**. Μπορείτε να δείτε την τεκμηρίωσή της στον σύνδεσμο: https://altair-viz.github.io/

Υπάρχουν διαθέσιμα κάποια βασικά αποσπάσματα κώδικα (code snippets) στη μεσαία επιλογή του μενού στα αριστερά των σημειωματαρίων. Περιμένουμε από εσάς να διαβάσετε την τεκμηρίωση και να καταλάβετε με ποιον τρόπο θα χρησιμοποιήσετε τη βιβλιοθήκη οπτικοποίησης. Η ενασχόλησή σας θα σας βοηθήσει τόσο στο Τμήμα 2 του δεύτερου μέρους της εργασίας, όσο και σε μελλοντική ενασχόλησή σας με ανάλυση δεδομένων.

Δείκτες/Μετρικές

Οι δείκτες του συνόλου δεδομένων World Bank είναι διαθέσιμοι και αναζητήσιμοι εδώ.

Είναι πιθανό να χρειαστεί να αναζητήσετε τους κωδικούς των δεικτών και τα πρότυπα κωδικών δεικτών (indicator codes - indicator code patterns) προκειμένου να εξάγετε τα απαραίτητα δεδομένα γι' αυτό το τμήμα της άσκησης. Όταν λοιπόν εντοπίσετε τον κατάλληλο δείκτη θα βρίσκεστε σε μια σελίδα με διεύθυνση της μορφής: `https://data.worldbank.org/indicator/XXXXXXXXX`. Τα X αντιστοιχούν στον `κωδικό του δείκτη (indicator_code)`. Για παράδειγμα στη σελίδα `https://data.worldbank.org/indicator/SH.XPD.CHEX.GD.ZS`, το `SH.XPD.CHEX.GD.ZS` είναι ο `κωδικός δείκτη` για τον οποίο θα κάνετε αναζητήσεις (επερωτήσεις) στο σύνολο δεδομένων.

Εναλλακτικά, μπορείτε να κάνετε ερωτήματα με λέξεις κλειδιά απευθείας στο BigQuery (είναι ευκολότερη διαδικασία για κάποιες απλούστερες γραφικές απεικονίσεις).

Πολλές από τις ερωτήσεις είναι *σκοπίμως* ανοιχτές αφήνοντάς σας να αποφασίσετε ποιοι είναι οι

καταλληλότεροι δείκτες (σπουδαία ικανότητα στην ανάλυση δεδομένων). Σημαντική παράμετρο στη διαμόρφωση και απάντηση ερωτημάτων είναι να σκέφτεστε τα "τυφλά σημεία" που έχουν οι δείκτες που θα χρησιμοποιήσετε. Για παράδειγμα, έστω ότι απεικονίζετε τα ευρώ του ξοδεύονται σε σχέση με το μορφωτικό επίπεδο για διάφορες χώρες. Δεν θα ήταν καλύτερο να μετρήσετε τις δαπάνες εν γένει ή το κεφάλαιο ως ποσοστό του ΑΕΠ; Ποια είναι τα αντισταθμιστικά οφέλη από τη χρήση των διαφορετικών δεικτών;

Ερώτηση 7 (3 μονάδες)

Αρχικά θα βρούμε κάτι στοιχειώδες - θα αναπαραστήσουμε γραφικά τον πληθυσμό της Ελλάδας ως διάγραμμα περιοχής σώρευσης (stacked area chart), για διάφορες ηλικιακές ομάδες που έχουν καταχωρηθεί στο σύνολο δεδομένων (0-14, 15-64, 65+). Ο x άξονας θα παριστάνει το έτος(year) και ο y τον πληθυσμό (population), για τις παραπάνω ηλικιακές ομάδες. Το άθροισμα όλων των περιοχών θα αντιπροσωπεύει το συνολικό πληθυσμό της Ελλάδας για ένα συγκεκριμένο έτος.

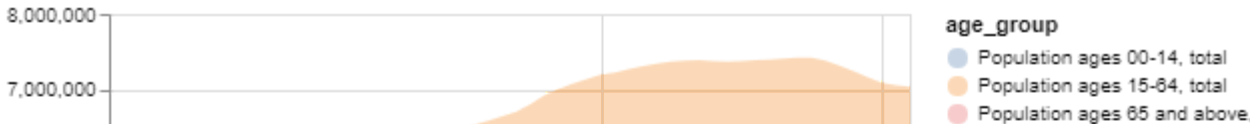
Υπόδειξη: Οι συναρτήσεις REGEX του BigQuery μπορεί να είναι χρήσιμες. Ελέγξτε εάν θελήσετε τη συνάρτηση regex που θα φτιάξετε εδώ πριν τη χρησιμοποιήσετε στο BigQuery για να βεβαιωθείτε ότι "δουλεύει".

```
%%bigquery --project $project_id q7

SELECT year, indicator_name AS age_group, value AS population
FROM `bigquery-public-data.world_bank_health_population.health_nutrition_population`
WHERE country_code = "GRC" AND (indicator_code = "SP.POP.0014.TO" OR indicator_code = "SP.P
```

12	1964	Population ages 00-14, total	2266476.0
13	1964	Population ages 15-64, total	5596606.0

14	1964	Population ages 65 and above, total	647347.0
15	1965	Population ages 65 and above, total	663531.0
16	1965	Population ages 15-64, total	5620822.0
17	1965	Population ages 00-14, total	2265980.0
18	1966	Population ages 65 and above, total	705077.0
19	1966	Population ages 00-14, total	2257475.0
20	1966	Population ages 15-64, total	5651099.0
21	1967	Population ages 65 and above, total	749353.0
22	1967	Population ages 00-14, total	2268793.0
23	1967	Population ages 15-64, total	5665941.0
24	1968	Population ages 00-14, total	2287046.0
25	1968	Population ages 65 and above, total	795322.0
26	1968	Population ages 15-64, total	5658396.0
27	1969	Population ages 65 and above, total	843361.0
28	1969	Population ages 15-64, total	5633417.0
29	1969	Population ages 00-14, total	2295987.0

```
alt.Chart(q7).mark_area(opacity=0.3).encode(
    x="year:T",
    y=alt.Y("population:Q", stack=None),
    color="age_group:N"
)
```



Ερώτηση 8 (3 μονάδες)

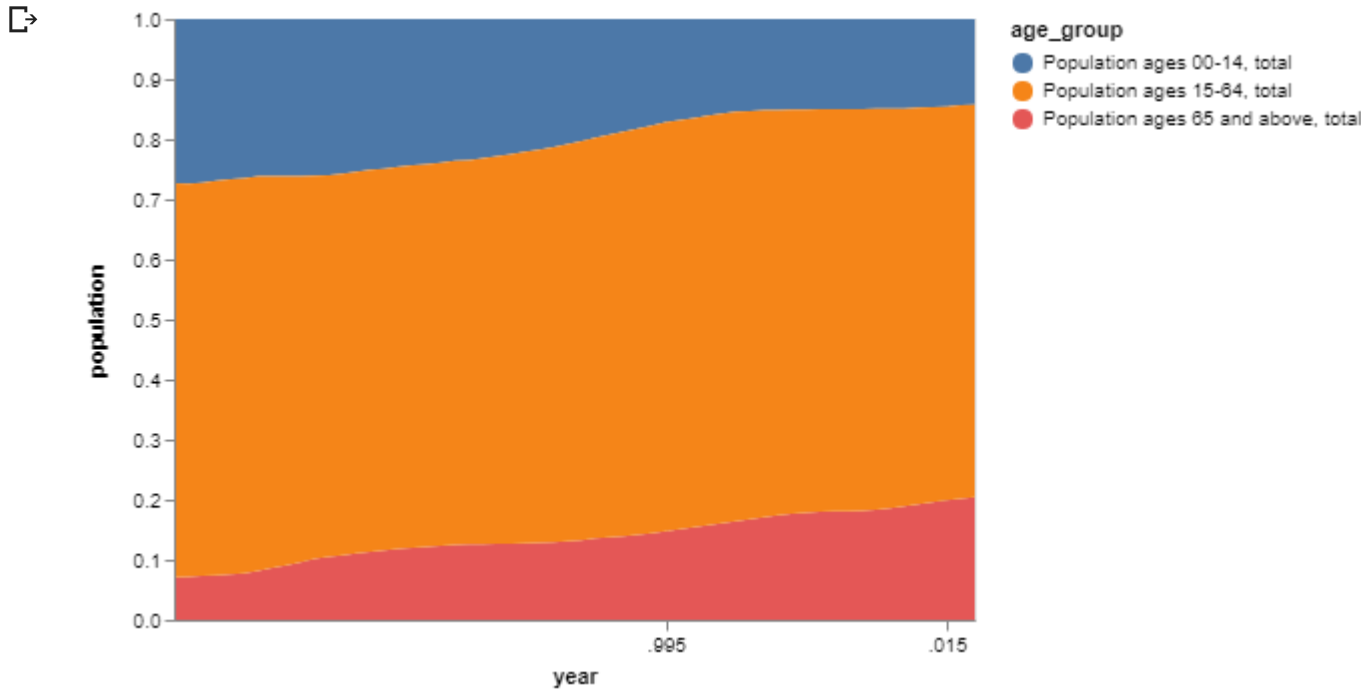
Στην Ελλάδα συνολικά έχουμε γήρανση ή ανανέωση του πληθυσμού; Φτιάξτε κανονικοποιημένο διάγραμμα περιοχής σώρευσης ώστε να "δείτε" την απάντηση στην ερώτηση!

```
%%bigquery --project $project_id q8

SELECT year, indicator_name AS age_group, value AS population
FROM `bigquery-public-data.world_bank_health_population.health_nutrition_population`
WHERE country_code = "GRC" AND (indicator_code = "SP.POP.0014.TO" OR indicator_code = "SP.P
```

	year	age_group	population
0	1960	Population ages 00-14, total	2278947.0

1	1960	Population ages 65 and above, total	587257.0
2	1960	Population ages 15-64, total	5465521.0
3	1961	Population ages 00-14, total	2304615.0
4	1961	Population ages 65 and above, total	601900.0
5	1961	Population ages 15-64, total	5491535.0
6	1962	Population ages 00-14, total	2302941.0
7	1962	Population ages 15-64, total	5528318.0
8	1962	Population ages 65 and above, total	616974.0

```
alt.Chart(q8).mark_area().encode(
    x="year:T",
    y=alt.Y("population:Q", stack="normalize"),
    color="age_group:N"
)
```



Ερώτηση 9 (4 μονάδες)

Ας φτιάξουμε μια γραφική παράσταση ακριβώς όπως το Gapminder ως απάντηση στην ερώτηση: "Ποιοι έχουν καλύτερη υγεία σε σχέση με τα χρήματα που ξοδεύουν;" Αναπαραστήσετε λοιπόν τα χρήματα που δαπανώνται στην υγεία (`money spent on healthcare`) ως προς το προσδόκιμο ζωής (`life expectancy`). "Παίξτε" με το Gapminder για να βρείτε τους κατάλληλους δείκτες (υπάρχουν διαφορετικές λύσεις) .

Φτιάξτε διάγραμμα φυσαλίδων (bubble plot) όπου το μέγεθος της φυσαλίδας αντιστοιχεί στον πληθυσμό της χώρας, το χρώμα της φυσαλίδας στη γεωγραφική περιοχή που ανήκει η χώρα και υπάρχει ολισθητής (slider) για αλλαγή στα έτη (σημείωση: διαλέξτε με λογικό τρόπο τα χρονικά διαστήματα). Συμπεριλάβετε επίσης έναν τρόπο για να δείχνετε ποια χώρα είναι κάθε φυσαλίδα

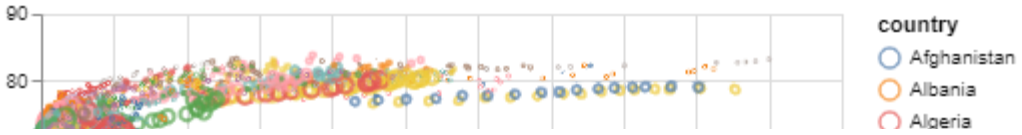
(ίσως ένα εργαλείο υποσημάτων - tooltip)

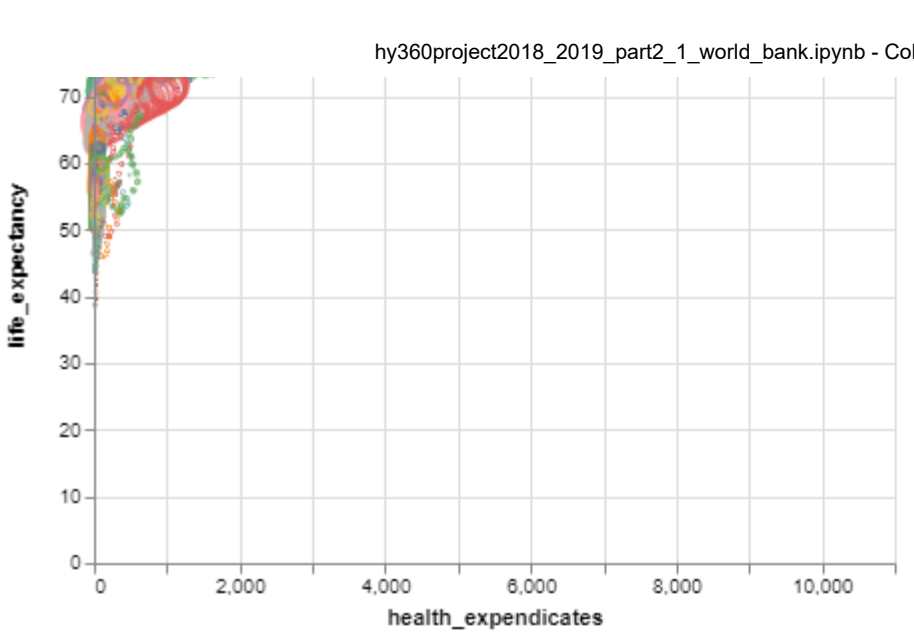
```
%%bigquery --project $project_id q9

SELECT country, t1.year AS year, health_expendicates, life_expectancy, value AS population
FROM
(SELECT country, t1.year AS year, health_expendicates, value AS life_expectancy
FROM
(SELECT country_name AS country, year, value AS health_expendicates
FROM `bigquery-public-data.world_bank_health_population.health_nutrition_population`
WHERE indicator_code = "SH.XPD.CHEX.PC.CD") AS t1
INNER JOIN `bigquery-public-data.world_bank_health_population.health_nutrition_population`
ON country = country_name AND t1.year = t2.year AND t2.indicator_code = "SP.DYN.LE00.IN") A
INNER JOIN `bigquery-public-data.world_bank_health_population.health_nutrition_population`
ON country = country_name AND t1.year = t2.year AND t2.indicator_code = "SP.POP.TOTL"
```

	country	year	health_expendicates	life_expectancy	population
0	Niger	2000	8.584913	49.874000	1.135297e+07

1	Tajikistan	2000	5.966178	65.485000	6.216205e+06
2	Latvia	2000	258.777923	70.314634	2.367550e+06
3	Lower middle income	2000	21.679349	62.670888	2.280235e+09
4	South Asia (IDA & IBRD)	2000	17.406992	62.865222	1.386626e+09
5	Costa Rica	2000	249.128670	77.448000	3.925443e+06
6	Belize	2000	132.615056	68.329000	2.473150e+05
7	Marshall Islands	2000	535.250407	65.239024	5.215900e+04
	Middle East & North				

```
slider = alt.binding_range(min=2000, max=2015, step=1)
select_year = alt.selection_single(name="year", fields=['year'], bind=slider)

alt.Chart(q9).mark_point().encode(
    x='health_expendicates',
    y='life_expectancy',
    size='population',
    color='country',
    tooltip=['country', 'life_expectancy', 'population']
).add_selection(select_year).transform_filter(select_year)
```





Angola
Antigua and Barbuda
Arab World
Argentina
Armenia
Australia
Austria
Azerbaijan
Bahamas, The
Bahrain
Bangladesh
Barbados
Belarus
Belgium
Belize
Benin
Bhutan
Bolivia
Bosnia and Herzegovina
Botswana
Brazil
Brunei Darussalam
Bulgaria
Burkina Faso
Burundi
Cabo Verde
Cambodia
Cameroon
Canada
Caribbean small states
Central African Republic
Central Europe and the Baltics
Chad
Chile
China
Colombia
Comoros
Congo, Dem. Rep.
Congo, Rep.
Costa Rica
Cote d'Ivoire
Croatia
Cyprus
Czech Republic
Denmark
Djibouti
Dominica
Dominican Republic
Early-demographic dividend
East Asia & Pacific
East Asia & Pacific (IDA & IBRD ...
East Asia & Pacific (excluding hig...
Ecuador
Egypt, Arab Rep.
El Salvador
Equatorial Guinea
Eritrea