HY360: Δεύτερο Μέρος Συνθετικής Εργασίας

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:

Μάριος Κωνσταντίνος Κωνσταντάκης, 3219.

Ιωακείμ Ορφέας Νικολουδάκης, 3682.

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 = 'groovy-analyst-227015'
```

```
# 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;

Τα δεδομένα στα σύνολα του World Bank έχουν οργανωθεί σε πίνακες που αντί να έχουν παραπάνω πλειάδες για διαφορετικούς τύπους γνωρισμάτων, έχουν τιμές γνωρισμάτων αποθηκευμένες σε γραμμές. Μοιάζει με πίνακα κατακερματισμού, καθώς βρίσκουμε την χώρα, μετά το γνώρισμα και στην συνέχεια την τιμή του.

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

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).

Πλεονεκτήματα OKV:

- Το κυριώτερο πλεονέκτημα της αποθήκης OKV είναι το ότι μπορούν να προστίθενται νέα χαρακτηριστικά στον πίνακα, δυναμικά και πολύ εύκολα. Απλά με την προσθήκη μιας γραμμής, χωρίς την εισαγωγή νέας πλειάδας στον πίνακα.
- Οι αποθήκες OKV δεν περιέχουν τιμές NULL. Οπότε αν έχουμε έναν πίνακα που οι περισσότερες πλειάδες του σε κάθε γραμμή ειναι κενές, τότε η αποθήκη OKV δεν θα σπαταλούσε χώρο σε κενές τιμές.
- Άμα το σχήμα της βάσης δεδομένων αλλάζει πολύ συχνά, δεν χρειάζεται να ξαναφτιαχτεί.

Μειονεκτήματα OKV:

- Οι αποθήκες OKV είναι λιγότερο γρήγορες απο τους κλασσικούς σχεσιακούς πίνακες, καθώς για να διαχειριστούμε μεγάλο όγκο δεδομένων χρειάζονται πολλαπλά join σε διάφορους πίνακες. Αυτό έχει ως αποτέλεσμα να μεγαλύτερη κατανάλωση πόρων, ιδιαίτερα αν οι πίνακες είναι ογκώδεις.
- Η σάρωση και η σύζευξη σε αυτές τις αποθήκες είναι πολύ πιο αργές σε αντίθεση με τους σχεσιακούς πίνακες. Ένας πίνακας σε OKV έχει συνήθως πολύ μεγάλο αριθμό γραμμών, οπότε η επίδωσή του είναι πολύ φτωχή. Εάν λάβουμε υπόψη ότι η πολυπλοκότητα της σάρωσης είναι O(N) και της σύζευξης O(N^2), είναι εύκολο να καταλάβει κάποιος ότι για έναν πολύ μεγάλο αριθμό N, η επίδωση των αλγορίθμων αυτών δεν θα είναι ικανοποιητική.
- Αν θέλουμε να αλλάξουμε την τιμή του κλειδιού θα πρέπει να το ψάξουμε και να το αλλάξουμε σε κάθε γραμμή από τις ατελείωτες γραμμές αυτού του πίνακα.

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.

Η εισαγωγή του πίνακα γνωρισμάτων διορθώνει την ανωμαλία ενημέρωσης. Δηλαδή, διευκολύνει την περίπτωση στην οποία χρειάζεται να αλλάξουμε την τιμή του pid χωρίς να χρειαστεί να αλλάξουμε τίποτα στον πίνακα δεδομένων. Όμως οι υπόλοιπες δυσκολίες που είχαμε πριν παραμένουν.

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;

Αυτή η λύση δεν ακολουθεί το μοντέλο της αποθήκης OKV επειδή σε κάθε γραμμή του πίνακα θα έχουμε πάντα μία τιμή null, το οποίο είναι ανεπιθύμητο.

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).

Μια άλλη σχεδίαση θα μπορούσε να είναι ο διαχωρισμός του παραπάνω πίνακα σε δύο πίνακες. Ο ένας πίνακας θα αποθηκεύει τους ακέραιους και ο άλλος τα αλφαριθμητικά. Δηλαδή, ένας πίνακας με τις πλειάδες {id, pid, string_value} και ένας με τις {id, pid, int_value}.

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;

Η δυσκολία στην σχεδίαση του προηγούμενου ερωτήματος είναι ότι για να πάρουμε όλη την πληροφορία για ένα συγκεκριμένο id, θα πρέπει να ψάξουμε σε δύο διαφορετικούς πίνακες. Από την μία πλεύρα, αν θέλουμε μόνο ακεραίους ή μόνο αλφαριθμητικά τα βρίσκουμε πιο γρήγορα, αλλά αν θέλουμε και τα δύο θα πρέπει να κάνουμε σύζευξη δύο πινάκων πολλών γραμμών.

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.

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;

Ο πίνακας γνωρισμάτων της world bank είναι ο world_bank_wdi.

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

Η περισσότερη πληροφορία για τα αντικείμενα περιέχεται στον πίνακα country_summary.

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

Το κλειδί του πίνακα health_nutrition_population είναι το indicator_name, 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)

Οντότητες: Country, Series, Indicator

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--(1,N)-- uses ---(1,N)-- Indicator, με γνωρίσματα πάνω στην σχέση uses τα year και value.

Country--(1,N)-- has --(1,N)-- Series, με γνώρισμα πάνω στην σχέση has τα year και description.

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)

Δώστε εδώ την απάντησή σας

Παράδειγμα σχεδίου πίνακα. Οι τιμές που θα περιέχει θα είναι δείγματα από το σύνολο world bank. Χρειαζόμαστε 2 με 3 γραμμές για να πάρουμε μια ιδέα σχετικά με το τι αποθηκεύεται σε κάθε στήλη:

Country

Country_name(κλειδί)	Country_code	Continent	Currency
Greece	GRC	Europe	Euro
United States	USA	North America	U.S. dollar

Country_Indicators

Country_name(ξένο κλειδί)	Country_code	indicator_name	indicator_code(κλειδί)	year	value
Greece	GRC	Population ages 00-14, total	SP.POP.0014.TO	1960	2278947.0
United States	USA	Population ages 00-14, total	SP.POP.0014.TO	1960	5.5442894E7

Country_Series

series_code(κλειδί)	Country_name(ξένο κλειδί)	Country_code	desription
SH.MMR.LEVE	Australia	AUS	Paid parental leave is available
SP.POP.0014.FE.ZS	China	CHN	Excluding Hong Kong SAR, Macao SAR, Taiwan.

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

Country:

```
Country_name -> Country_code
Country_name -> Continent
Country_name -> Currency
```

country_indicators:

```
Country_name -> country_code
Indicator_code -> Indicator_name
```

```
Indicator_code, year -> value
```

country_series:

```
Series_code -> description
Country_name -> country_code
```

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

Δώστε εδώ την απάντησή σας

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)

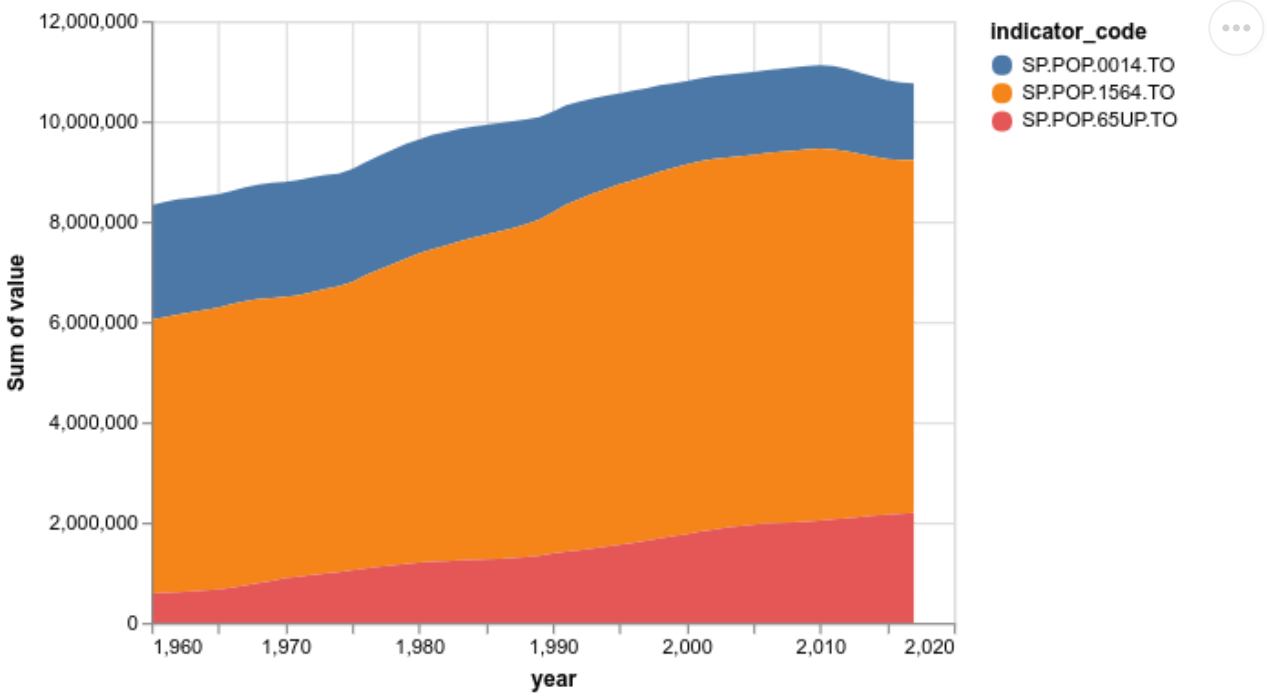
Αρχικά θα βρούμε κάτι στοιχειώδες - θα αναπαραστήσουμε γραφικά τον πληθυσμό της Ελλάδας ως διάγραμμα περιοχής σώρευσης (stacked area chart), για διάφορες ηλικιακές ομάδες που έχουν καταχωρηθεί στο σύνολο δεδομένων (0-14, 15-64, 65+). Ο x άξονας θα παριστάνει το `έτος (year)` και ο y τον `πληθυσμό (population)`,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

```
%%bigquery --project groovy-analyst-227015  q7

SELECT year, value, indicator_code
FROM `bigquery-public-data.world_bank_health_population.health_nutrition_populatio
WHERE country_name = "Greece" AND
(indicator_code = "SP.POP.0014.TO" OR indicator_code = "SP.POP.1564.TO" OR indicat
ORDER BY year, indicator_code
```

	year	value	indicator_code
0	1960	2278947.0	SP.POP.0014.TO
1	1960	5465521.0	SP.POP.1564.TO
2	1960	587257.0	SP.POP.65UP.TO

```
import pandas as pd
import altair as alt

source = q7.value

alt.Chart(q7).mark_area().encode(
x = 'year',
y = 'sum(value)',
color = "indicator_code")
```



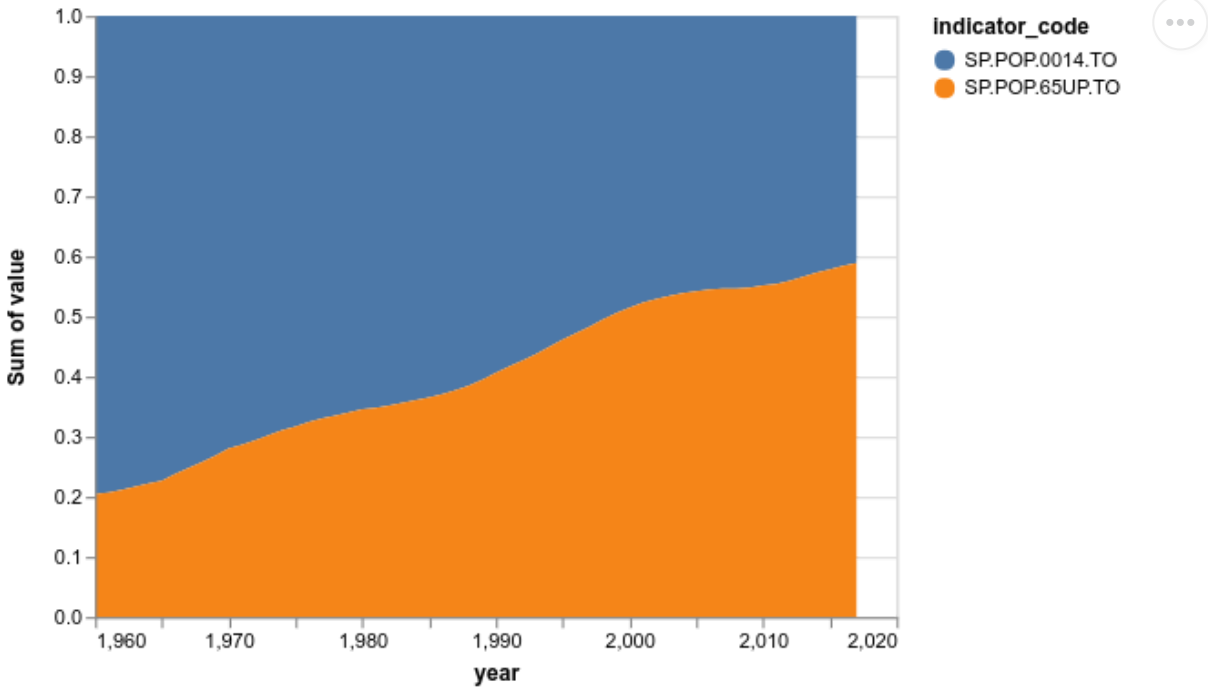
Ερώτηση 8 (3 μονάδες)

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

```
%%bigquery --project groovy-analyst-227015 q8
#SELECT year, ROUND(value,2),indicator_code
#FROM `bigquery-public-data.world_bank_health_population.health_nutrition_populati
#WHERE country_name = "Greece" AND (indicator_code = "SP.POP.65UP.TO.ZS" OR indica
#ORDER BY year

SELECT year, value, indicator_code
FROM `bigquery-public-data.world_bank_health_population.health_nutrition_populatio
WHERE country_name = "Greece" AND
(indicator_code = "SP.POP.0014.TO" OR indicator_code = "SP.POP.65UP.TO")
ORDER BY year, indicator_code
```

	year	value	indicator_code
0	1960	2278947.0	SP.POP.0014.TO
1	1960	587257.0	SP.POP.65UP.TO
2	1961	2304615.0	SP.POP.0014.TO
3	1961	601900.0	SP.POP.65UP.TO
4	1962	2302941.0	SP.POP.0014.TO
5	1962	616974.0	SP.POP.65UP.TO
6	1963	2282633.0	SP.POP.0014.TO
7	1963	631972.0	SP.POP.65UP.TO
8	1964	2266476.0	SP.POP.0014.TO
9	1964	647347.0	SP.POP.65UP.TO
10	1965	2265980.0	SP.POP.0014.TO
11	1965	663531.0	SP.POP.65UP.TO
12	1966	2257475.0	SP.POP.0014.TO
13	1966	705077.0	SP.POP.65UP.TO
14	1967	2268793.0	SP.POP.0014.TO
15	1967	749353.0	SP.POP.65UP.TO
16	1968	2287046.0	SP.POP.0014.TO
17	1968	795322.0	SP.POP.65UP.TO
18	1969	2295987.0	SP.POP.0014.TO
19	1969	843361.0	SP.POP.65UP.TO
20	1970	2292326.0	SP.POP.0014.TO
21	1970	895344.0	SP.POP.65UP.TO
22	1971	2295990.0	SP.POP.0014.TO
23	1971	924012.0	SP.POP.65UP.TO
24	1972	2290101.0	SP.POP.0014.TO
25	1972	955997.0	SP.POP.65UP.TO
26	1973	2270885.0	SP.POP.0014.TO
27	1973	986358.0	SP.POP.65UP.TO
28	1974	2251507.0	SP.POP.0014.TO

```
import pandas as pd
import altair as alt

alt.Chart(q8).mark_area().encode(
x = 'year',
y = alt.Y('sum(value)', stack = "normalize"),
color = "indicator_code")
```



Ερώτηση 9 (4 μονάδες)

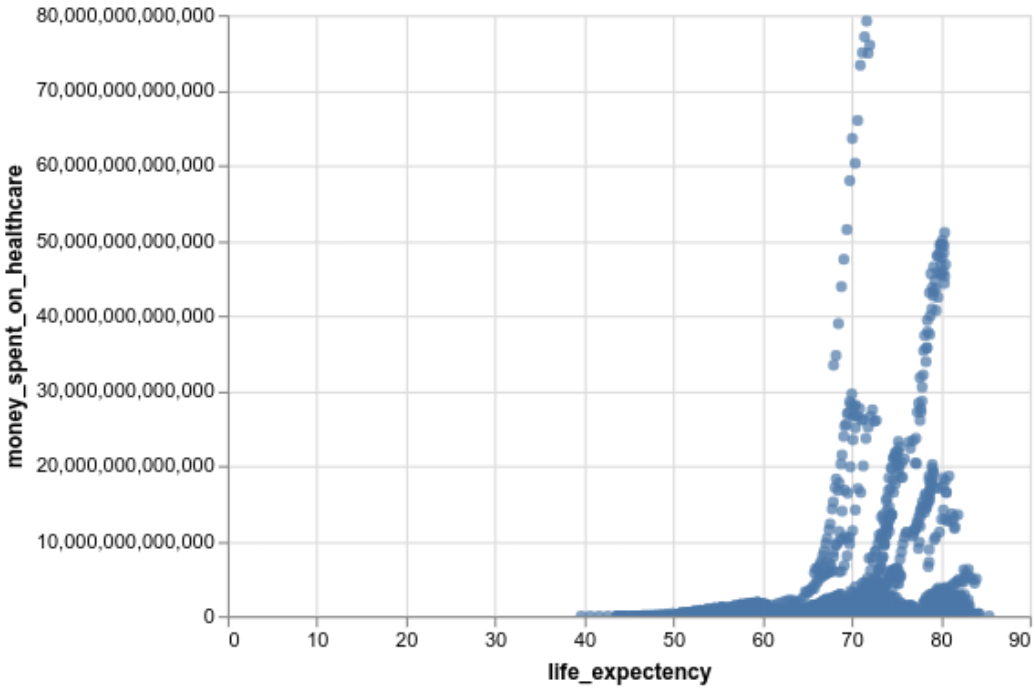
Ας φτιάξουμε μια γραφική παράσταση ακριβώς όπως το Gapminder ως απάντηση στην ερώτηση: "Ποιοι έχουν καλύτερη υγεία σε σχέση με τα χρήματα που ξοδεύουν;" Αναπαραστήσετε λοιπόν τα χρήματα που δαπανώνται στην υγεία (money spent on healthcare) ως προς το προσδόκιμο ζωής (life expectancy). "Παίξτε" με το 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 υπομνήσεων - tooltip).

```
%%bigquery --project groovy-analyst-227015 q9

SELECT ind1.country_name,  ind1.year, ind1.value AS life_expectency, ind2.money_sp
FROM `bigquery-public-data.world_bank_wdi.indicators_data` ind1
JOIN (SELECT country_name, indicator_name, year, value AS money_spent_on_healthcar
      FROM  `bigquery-public-data.world_bank_wdi.indicators_data`
      WHERE indicator_code = "NY.GDP.MKTP.CD" AND year > 2000)ind2
ON  (ind1.country_name = ind2.country_name AND ind1.year = ind2.year)
WHERE indicator_code = "SP.DYN.LE00.IN" AND ind1.year > 2000
order by country_name, year
```

	country_name	year	life_expectency	money_spent_on_healthcare
0	Afghanistan	2002	56.637	4.055180e+09
1	Afghanistan	2003	57.250	4.515559e+09
2	Afghanistan	2004	57.875	5.226779e+09
3	Afghanistan	2005	58.500	6.209138e+09
4	Afghanistan	2006	59.110	6.971286e+09
5	Afghanistan	2007	59.694	9.747880e+09
6	Afghanistan	2008	60.243	1.010923e+10
7	Afghanistan	2009	60.754	1.243909e+10
8	Afghanistan	2010	61.226	1.585657e+10
9	Afghanistan	2011	61.666	1.780429e+10
10	Afghanistan	2012	62.086	1.990732e+10
11	Afghanistan	2013	62.494	2.056107e+10
12	Afghanistan	2014	62.895	2.048489e+10
13	Afghanistan	2015	63.288	1.990711e+10
14	Afghanistan	2016	63.673	1.904636e+10
15	Albania	2001	74.286	3.922101e+09
16	Albania	2002	74.575	4.348068e+09
17	Albania	2003	74.820	5.611496e+09
18	Albania	2004	75.028	7.184686e+09
19	Albania	2005	75.217	8.052074e+09
20	Albania	2006	75.418	8.896073e+09
21	Albania	2007	75.656	1.067732e+10
22	Albania	2008	75.943	1.288135e+10
23	Albania	2009	76.281	1.204421e+10
24	Albania	2010	76.652	1.192696e+10

```
import pandas as pd
import altair as alt

slider = alt.binding_range(min=2001, max=2018, step=1)
years = q9.year
select_year = alt.selection_single(name="year", fields=['years'], bind=slider)


alt.Chart(q9).mark_circle().encode(
x = 'life_expectency:Q',
y = 'money_spent_on_healthcare:Q',
tooltip = [ 'country_name' , 'year', 'life_expectency' , 'money_spent_on_healthcar
  ).add_selection(select_year).interactive()
```


80,000,000,000,000
70,000,000,000,000
60,000,000,000,000
50,000,000,000,000
40,000,000,000,000
30,000,000,000,000
20,000,000,000,000
10,000,000,000,000
0
money_spent_on_healthcare
0
10
20
30
40
50
60
70
80
90
life_expectency

year_years