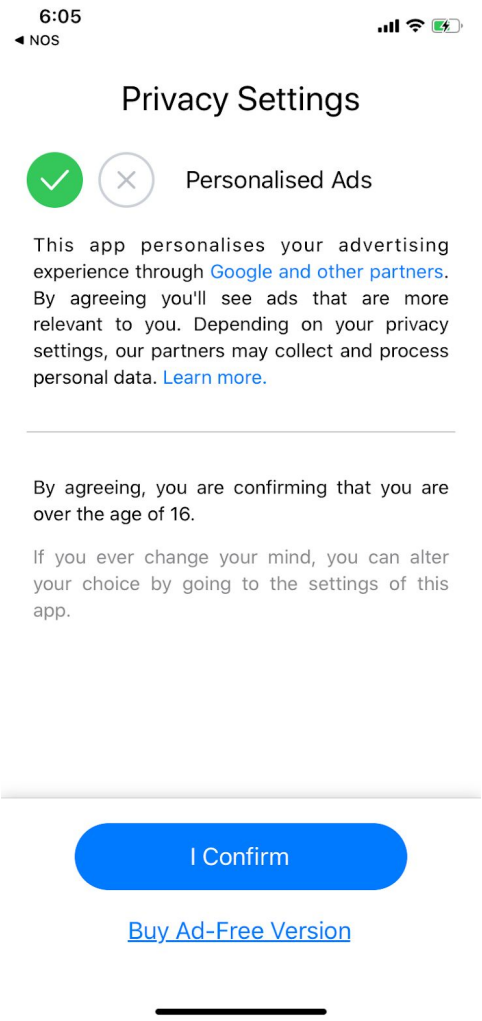
SQL Test

For this test you will make use of some data available in this folder. You can set up a database by running a docker instance. See the installation steps in the README.md file. Alternatively, you can create your own local database named ‘consent’ and run the dump script which you can find in `init/consent_log_2_sql`.

The tables contain analytics about some pop-ups that we have in our apps that allow users to provide consent for the GDPR and CCPA. We are not allowed to log an identifier before the user has given consent, as this constitutes personal data. Therefore we are counting the number of ‘opens’, ‘closes’ and custom events anonymously in order to have an idea of what percentage of users finishes the consent pop up.

See the GDPR popup in the image below.



Questions

Create a query for every question. Put your answers in a new document.

1. Show all events in dialog_show with regulation_id '1'.
2. Show all events in dialog_show with regulation_id '1', from March 2020.
3. Count the number of events from March 2020 in dialog_show per regulation.
4. Show all unique events from dialog_event in March 2020 ordered alphabetically.
5. Show top 10 locales from dialog_close with the most close counts in March 2020.
6. Show bottom 10 locales from dialog_close where the summed close counts in March 2020 were at least 100 or more.
7. Show all events from dialog_event, but instead of app_id, regulation_id, origin_id, flow_id and run_counter_id show the human readable names found in the corresponding tables.
8. (Bonus) Show the summed counts per run_counter ordered from high to low, for all run_counters but 'old'. Combine 'firstRun', 'secondRun' and 'thirdRun' under one new group called 'firstToThirdRunCount'. (Hint: Use case statement)