



1. Untuk setiap movie, tampilkan id movie, judul movie, tahun rilis, dan jumlah episode nya, lalu urutkan dari tahun rilis terbaru hingga terlama. Untuk movie yang tidak memiliki episode tidak perlu ditampilkan.
2. Tampilkan jumlah episode dari movie "Cells at Work" yang telah ditonton oleh user bernama "Andrew". Jika Andrew menonton episode yang sama lebih dari sekali maka tetap dihitung 1.
3. Tampilkan semua judul movie, dimana memiliki episode yang telah ditonton lebih dari 1 kali. Contoh: movie "Cells at Work" memiliki episode yang telah ditonton lebih dari 1 kali (pada season 2 episode 1).
4. Tampilkan nama dan email dari user yang durasi menontonnya selalu lebih dari 1 jam.
5. Tampilkan semua informasi movie (judul movie, judul episode, nama genre) mana saja yang dapat ditonton oleh user bernama 'Andrew', 'Nobita', dan 'Monalisa' sesuai usia mereka. Perhatikan bahwa movie yang tidak memiliki episode yang memenuhi requirement ini juga tetap ditampilkan, dengan judul episode nya bernilai NULL

Petunjuk Anda dapat menggunakan fungsi `date_part('year', TANGGAL)` untuk mendapatkan tahun dari suatu variabel TANGGAL yang bertipe date.

6. Buatlah stored procedure **countNumberOfViewers** untuk menghitung jumlah user yang telah menonton movie untuk suatu season dan episode tertentu. Stored procedure ini menerima 3 buah argumen, yaitu: judul movie, nomor season, dan nomor episode. Asumsi nilai judul movie, nomor season dan nomor episode yang diinputkan pada fungsi ini pasti bukan NULL. Contoh: jumlah viewer (user yang telah menonton) film Kungfu Panda The Dragon Knight season 1 episode 2 adalah 3 orang. Perhatikan bahwa jika seorang user yang sama menonton suatu episode berkali-kali maka akan tetap dihitung sebagai 1 viewer.

Setelah stored procedure dibuat, lakukan pengujian dengan memanggil:

```
SELECT countNumberOfViewers('Kungfu Panda the Dragon Knight',  
1, 2);
```

7. Buatlah stored procedure **recommendedMoviePercentage** untuk menghitung persentase jumlah movie direkomendasikan (`recommended="YES"`) oleh user yang sudah menonton movie tertentu. Stored procedure ini menerima 1 buah argumen, yaitu judul movie. Contoh: pemanggilan **recommendedMoviePercentage('Cells at Work')** akan mengembalikan nilai 66 karena ada 2 nilai rekomendasi "YES" dari total 3 nilai rekomendasi yang diberikan untuk movie "Cells at Work", sehingga hasilnya menjadi $100\% * (\frac{2}{3}) = 66\%$. Setelah stored procedure dibuat, lakukan pengujian dengan memanggil:

```
SELECT recommendedMoviePercentage('Cells at Work');
```

8. Buatlah stored procedure & trigger **checkWatchingDuration** untuk memastikan bahwa durasi menonton suatu episode movie dari seorang pengguna tidak mungkin lebih besar dari durasi episode itu sendiri. Perhatikan event apa saja yang perlu mengaktifkan trigger yang Anda buat. Setelah stored procedure & trigger dibuat, lakukan pengujian untuk satu perintah SQL berikut ini:

```
insert into watch_history values (11, 'mipo@gmail.com', 10, 1,  
3, '2020-10-28 10:00', '03:00:00', 'YES');
```