

Instituto Superior Técnico

Projeto de Base de Dados,

Parte 4

Ruben Correia nº85909

Raquel Pereira nº76171

Joaquim Esteves nº77020

Grupo nº 99

Turno: Quinta-feira 12:30h – 14:00h

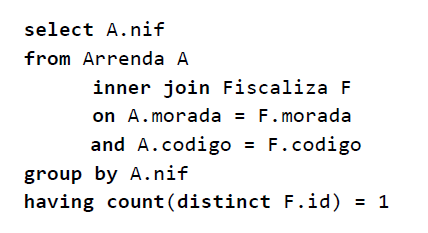
Esforço: 72h

Prof. Gabriel Pestana

**Índices**

**a)**

1.



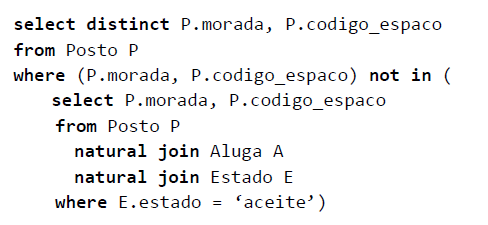
Índice composto Arrenda (morada, código, nif) pois é útil quando temos uma combinação de campos, como no caso desta query o select ser feito de acordo com os campos morada, código e nif. Não é necessário um índice para aceder ao id da tabela Fiscaliza pois este é o primary key da mesma logo já é um índice e este vai ter de percorrer a tabela toda de qualquer maneira para contar logo não vai fazer diferença criar um índice para o count.

**2.**

Índice comporto Estado (numero, estado)

Índice comporto Posto (morada, codigo\_espaco)

No caso do índice composto estado seria melhor este ser um índice do tipo HASH pois é mais eficiente para comparações de igualdade como o = mas como o nosso MySQL só suport índices do tipo BTREE este não pode ser implementado.



b)

query\_1

0.010474

query\_2

0.04645200

**Data Warehouse**

1.

**local\_dimension (DT)**

local\_id (PK)

local\_building

local\_space

local\_post

payed\_amount

**user\_dimension (DT)**

user\_nif (PK)

user\_telephone

user\_name

**reservation\_fact (FT)**

reservation\_id (PK)

local\_id (PK, FK)

date\_id (PK, FK)

user\_nif (PK, FK)

time\_id (PK, FK)

status\_id (PK, FK)

duration\_in\_days

**time\_dimension (DT)**

time\_id (PK)

time\_hour

time\_minute

**date\_dimension (DT)**

date\_id (PK)

date\_day

date\_week

date\_month\_number

date\_semester

date\_year

**status\_dimension (DT)**

status\_id (PK)

status

**Esquema da estrela:**

CREATE TABLE user\_dimension (

user\_nif varchar (9) NOT NULL UNIQUE,

user\_name varchar (80) NOT NULL,

user\_telephone varchar (26) NOT NULL,

primary key(user\_nif));

CREATE TABLE local\_dimension (

local\_id integer NOT NULL UNIQUE,

local\_building varchar (255) NOT NULL,

local\_post varchar (255) DEFAULT 'NO POSTO AVAILABLE',

local\_space varchar (255) DEFAULT 'NO ESPACO AVAILABLE',

primary key (local\_id));

CREATE TABLE time\_dimension (

time\_id integer NOT NULL UNIQUE,

time\_hour integer,

time\_minute integer,

primary key (time\_id));

CREATE TABLE date\_dimension (

date\_id integer NOT NULL UNIQUE,

date\_day integer,

date\_week integer,

date\_month varchar (255),

date\_year integer,

date\_semester integer,

primary key(date\_id));

CREATE TABLE status\_dimension (

status\_id integer NOT NULL UNIQUE,

status varchar (255));

CREATE TABLE reservation\_fact (

reservation\_id varchar (255) NOT NULL UNIQUE,

user\_nif varchar (9) NOT NULL,

local\_id integer NOT NULL,

time\_id integer NOT NULL,

date\_id integer NOT NULL,

status varchar (255) NOT NULL,

payed\_amount numeric (19,4) NOT NULL,

duration\_in\_days integer NOT NULL,

primary key (reservation\_id, user\_nif, local\_id, time\_id, date\_id),

foreign key (user\_nif) references user\_dimension(user\_nif) ON DELETE CASCADE,

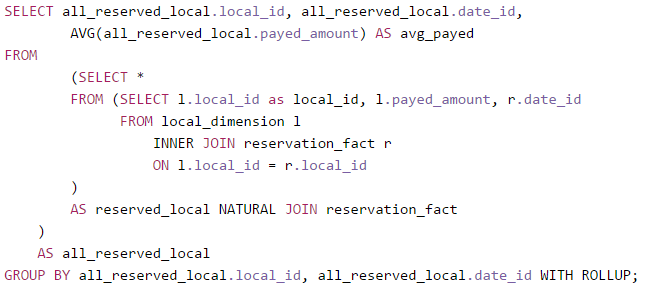
foreign key (local\_id) references local\_dimension(local\_id) ON DELETE CASCADE,

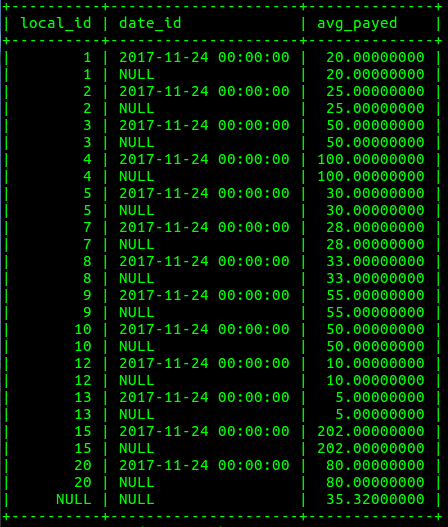
foreign key (time\_id) references time\_dimension(time\_id) ON DELETE CASCADE,

foreign key (date\_id) references date\_dimension(date\_id) ON DELETE CASCADE);

(em anexo estão os populates das tabelas)

**2.** A consulta OLAP, para obter o cubo com o valor médio pago sobre a dimensões localização(local\_id) e data(date\_id), é a seguinte:





**Fig.1 –** Cubo obtido, como exemplo, a partir da nossa consulta OLAP