

Previsão de Carga

O número de linhas estimado terá um valor relativo, poderá ser: very low, low, medium, high, very high, super high, aroundRows(table).

Tabela	Número de linhas	Justificação
Evento	high	Principal atração do site
Utilizador	medium	A população do site será inferior no nosso ver aos Eventos
Sondagem	aroundRows(Evento)	É provável cada Evento ter mais ou menos uma Sondagem
Opcao	aroundRows(4*Evento*1)	Por volta de 4 opções numa sondagem
Comentario	very high	Comunidade bastante ativa
Pais	very low	Por volta de 200 linhas
Seguidor	very high	Cada utilizador pode ser seguido por milhares de pessoas
Participacao	very high	Contabilização de quem vai ao evento
Convite	super high	É um método de publicidade
Anfitriao	high/very high	Existe pelo menos um num evento, e geralmente não há muitos
UtilizadorOpcao	high	Será menos que Opcao e mais que sondagem
ComentarioVoto	super high	Mais entradas que Comentario
Administrador	very low	Numero reduzido de Administradores
Album	aroundRows(Evento)	Cada Evento terá mais ou menos um Album
Imagem	very high	Existirá com muito probabilidade muitas fotos partilhadas

Selects

[g22_selects.sql](#)

```
-- Get login Information of an "Utilizador" by email
SELECT *
FROM Utilizador
WHERE email = 'drumond@gmail.com';
--

-- Get login Information of an "Utilizador" by username
SELECT *
FROM Utilizador
WHERE username = 'blackhouse';
--

-- Find website events, FULL TEXT SEARCH
CREATE OR REPLACE VIEW Event_view_fts AS
SELECT Evento.idEvento AS id,
       Evento.titulo AS title,
       Evento.descricao AS descricao,
       setweight(to_tsvector('english', Evento.titulo), 'A') ||
```

```

        setweight(to_tsvector('english', COALESCE(Evento.descricao,
        '')), 'B') ||
        setweight(to_tsvector('simple', Evento.localizacao), 'C') ||
        setweight(to_tsvector('simple', string_agg(Utilizador.username,
        ' ')), 'B') AS document
FROM Evento
JOIN Anfitriao ON Anfitriao.idEvento = Evento.idEvento
JOIN Utilizador ON Utilizador.idUtilizador = Anfitriao.IdAnfitriao
GROUP BY Evento.idEvento;

SELECT id, title, descricao
FROM Event_view_fts
WHERE Event_view_fts.document @@ to_tsquery('english', 'water')
ORDER BY ts_rank(Event_view_fts.document, plainto_tsquery('english
water')) DESC;
--

-- Find website events of a given "Utilizador", FULL TEXT SEARCH
CREATE OR REPLACE VIEW Event_view_user_fts AS
SELECT Evento.idEvento AS id,
        Evento.titulo AS title,
        Evento.descricao AS descricao,
        setweight(to_tsvector('english', Evento.titulo), 'A') ||
        setweight(to_tsvector('english', COALESCE(Evento.descricao,
        '')), 'B') ||
        setweight(to_tsvector('simple', Evento.localizacao), 'C') AS
Document
FROM Evento
JOIN Anfitriao ON Anfitriao.idEvento = Evento.idEvento
WHERE
        Anfitriao.idAnfitriao = (SELECT Utilizador.idUtilizador FROM
Utilizador WHERE Utilizador.username = 'avc')
GROUP BY Evento.idEvento;

SELECT id, title, descricao
FROM Event_view_user_fts
WHERE Event_view_user_fts.document @@ to_tsquery('MATDSL')
ORDER BY ts_rank(Event_view_user_fts.document, plainto_tsquery('english
MATDSL')) DESC;
--

-- Find the comments concerning an "Evento" and the votes
SELECT Results.idComentario, Results.idComentador, Results.username,
Results.texto, Results.data, Results.idComentarioPai,
json_object_agg(Results.positivo, json_build_object('voters',
Results.voters, 'votes', Results.count)) AS votes
FROM (
        SELECT Comentario.idComentario, Comentario.idComentador, (SELECT
Utilizador.username FROM Utilizador WHERE Utilizador.idUtilizador =
Comentario.idComentario) AS username, Comentario.texto,
Comentario.data, json_agg(json_build_object('id',

```

```
Utilizador.idUtilizador, 'username', Utilizador.username)) AS voters,
Comentario.idComentarioPai, positivo, COUNT(positivo) AS COUNT
FROM Comentario
JOIN ComentarioVoto ON ComentarioVoto.idComentario =
Comentario.idComentario
JOIN Utilizador ON Utilizador.idUtilizador = ComentarioVoto.idVotante
WHERE Comentario.idEvento = 1
GROUP BY Comentario.idComentario, ComentarioVoto.Positivo
) AS Results
GROUP BY Results.idComentario, Results.idComentador, Results.username,
Results.texto, Results.data, Results.idComentarioPai;
--

-- Get the albums and images of an "Evento"
SELECT Album.idAlbum, Album.nome, Album.descricao,
json_agg(json_build_object('id', Imagem.IdImagem, 'caminho',
Imagem.caminho, 'data', Imagem.data))
FROM Album
JOIN Imagem ON Imagem.idAlbum = Album.idAlbum
WHERE Album.idEvento = 2
GROUP BY Album.idAlbum;
--

-- Get the "Sondagem" and its "opcao"
SELECT Sondagem.IdSondagem, Sondagem.descricao, Sondagem.data,
Sondagem.escolhaMultipla, json_agg(json_build_object('id',
Opcao.idOpcao, 'descricao', Opcao.descricao))
FROM Sondagem
JOIN Opcao ON Opcao.idSondagem = Sondagem.idSondagem
WHERE Sondagem.idEvento = 1
GROUP BY Sondagem.idSondagem;
--

-- Get the current results of a "Sondagem"
SELECT json_agg(ResultsById.ResultsById) AS SondagemResults
FROM (
    SELECT json_build_object('id', Opcao.idOpcao, 'votes',
COUNT(Opcao.idOpcao)) AS ResultsById
    FROM Opcao
    JOIN UtilizadorOpcao ON Opcao.idOpcao = UtilizadorOpcao.idOpcao
    WHERE Opcao.idSondagem = 1
    GROUP BY Opcao.idOpcao
) AS ResultsById;
--

-- Get the Participants of an "Evento"
SELECT Utilizador.idUtilizador, Utilizador.username,
Participacao.classificacao, Participacao.comentario
FROM Participacao
JOIN Utilizador ON Participacao.IdParticipante =
Utilizador.idUtilizador
```

```
WHERE Participacao.idEvento = 1;
--

-- Get the people that are Participants of an "Evento" that I follow
SELECT Utilizador.idUtilizador, Utilizador.username
FROM Participacao
JOIN Seguidor ON Participacao.IdParticipante IN (SELECT
Seguidor.idSeguido FROM Seguidor WHERE Seguidor.idSeguido = 7)
JOIN Utilizador ON Utilizador.IdUtilizador =
Participacao.IdParticipante AND Seguidor.idSeguido =
Participacao.IdParticipante
WHERE Participacao.idEvento = 2;
--

-- Get the top 10 (if exists at least 10) upcoming public Events
SELECT * FROM Evento
WHERE publico = TRUE
ORDER BY dataInicio ASC
LIMIT 10;
--

-- Get the top 10 (if exists at least 10) with most participants
SELECT E.idEvento, E.titulo, E.capa, E.descricao, E.localizacao,
E.dataInicio, E.duracao, E.publico, P.Numero_de_Participantes
FROM Evento E
INNER JOIN
(
    SELECT idEvento, COUNT(idEvento) AS Numero_de_Participantes
    FROM Participacao
    GROUP BY idEvento
) P ON E.idEvento = P.idEvento
ORDER BY Numero_de_Participantes DESC
LIMIT 10;
--
```

Indexes

Os índices nas chaves estrangeiras devem ser criados quando:

- A tabela que contém a chave externa é grande e a referenciada está sujeita a *updates* e *deletes*
- A referenciada é grande e a chave externa é usada em *joins*
- A chave externa é usada em *WHEREs* numa tabela grande

[g22_indexes.sql](#)

```
-- Index for login
```

```
CREATE UNIQUE INDEX
utilizador_username_index ON Utilizador(username);

-- Index for login
CREATE UNIQUE INDEX
utilizador_email_index ON Utilizador(email);

-- Full text search indexes for Evento, these are always used together
CREATE INDEX fts_evento_index
ON Evento
USING
gin((
    setweight(to_tsvector('english', titulo), 'A') ||
    setweight(to_tsvector('english', descricao), 'B') ||
    setweight(to_tsvector('english', localizacao), 'B')
));
```

Updates & Deletes

[updatesDeletes.sql](#)

```
-- UPDATES
UPDATE Evento
SET descricao = 'A festa e fixe'
WHERE idEvento = 1;

UPDATE Evento
SET dataInicio = '2016-08-04'
WHERE idEvento = 3;

UPDATE Utilizador
SET password = 'arrozdechocolate'
WHERE idUtilizador = 1;

UPDATE Utilizador
SET foto = 'top.png'
WHERE idUtilizador = 2;

UPDATE Sondagem
SET escolhaMultipla = 'TRUE'
WHERE idSondagem = 1;

UPDATE UtilizadorOpcao
SET IdOpcao = 5,
WHERE IdUtilizador = 1 AND IdOpcao = 3;

UPDATE Convite
```

```
SET resposta = 'TRUE'
WHERE idEvento = 1 AND idConvidado = 2;

-- DELETES
DELETE FROM Comentario
WHERE IdComentario = 1;

DELETE FROM Opcao
WHERE idOpcao = 10;

DELETE FROM Seguidor
WHERE IdSeguidor = 2 AND IdSeguido = 1;

DELETE FROM UtilizadorOpcao
WHERE IdUtilizador = 2 AND IdOpcao = 5;

DELETE FROM ComentarioVoto
WHERE IdComentario = 2 AND IdVotante = 2;

DELETE FROM Imagem
WHERE IdImagem = 5;
```

Triggers

[g22_triggers.sql](#)

```
-- Prevent Rating if user did not participate or the event is not
completed
DROP TRIGGER IF EXISTS CanRate ON Participacao;

CREATE OR REPLACE FUNCTION trigger_canRate() RETURNS TRIGGER AS $$
BEGIN
    IF NOT check_participation(NEW.idParticipante) THEN
        RAISE EXCEPTION 'Current user did not participate in the event';
    END IF;
    IF NOT check_event_complete(NEW.idEvento) THEN
        RAISE EXCEPTION 'Event is not complete yet';
    END IF;
    RETURN NEW;
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER CanRate BEFORE INSERT ON Participacao
FOR EACH ROW
EXECUTE PROCEDURE trigger_canRate();

-- Prevent Users from following themselves
```

```
DROP TRIGGER IF EXISTS CanFollow ON Seguidor;

CREATE OR REPLACE FUNCTION trigger_canFollow() RETURNS TRIGGER AS $$
BEGIN
    IF (NEW.IdSeguidor = NEW.IdSeguido) THEN
        RAISE EXCEPTION 'Users cannot follow themselves';
    END IF;
    RETURN NEW;
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER CanFollow BEFORE INSERT ON Seguidor
FOR EACH ROW
EXECUTE PROCEDURE trigger_canFollow();

-- Prevent Users from voting on their own comments
DROP TRIGGER IF EXISTS CanVote ON ComentarioVoto;

CREATE OR REPLACE FUNCTION trigger_canVote() RETURNS TRIGGER AS $$
BEGIN
    IF NOT check_vote(NEW.IdComentario, NEW.IdVotante) THEN
        RAISE EXCEPTION 'Users cannot vote on their own comments';
    END IF;
    RETURN NEW;
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER CanVote BEFORE INSERT ON ComentarioVoto
FOR EACH ROW
EXECUTE PROCEDURE trigger_canVote();
```

User-defined Functions

[g22_functions.sql](#)

```
-- Check if event is completed
CREATE OR REPLACE FUNCTION check_event_complete(INTEGER) RETURNS
BOOLEAN AS $$
    SELECT EXISTS (
        SELECT idEvento, dataInicio, duracao FROM Evento WHERE idEvento =
$1 AND dataInicio + (duracao * INTERVAL '1 second') >=
CURRENT_TIMESTAMP
    );
$$ LANGUAGE SQL;

-- Check if user participated in event
CREATE OR REPLACE FUNCTION check_participation(INTEGER) RETURNS BOOLEAN
```

```
AS $$
    SELECT EXISTS (
        SELECT Evento.idEvento, idParticipante FROM Evento, Participacao
WHERE Evento.idEvento = Participacao.idParticipante AND
Participacao.idParticipante = $1
    );
$$ LANGUAGE SQL;

-- Check if user can vote in the comment
CREATE OR REPLACE FUNCTION check_vote(INTEGER, INTEGER) RETURNS BOOLEAN
AS $$
    SELECT EXISTS (
        SELECT IdComentario, IdComentador FROM Comentario WHERE
Comentario.IdComentario = $1 AND Comentario.IdComentador = $2
    );
$$ LANGUAGE SQL;
```

[\[EventBook\]](#)

From:

<http://lbaw.fe.up.pt/201516/> - **L B A W :: WORK**

Permanent link:

<http://lbaw.fe.up.pt/201516/doku.php/lbaw1522/proj/a7>

Last update: **2016/04/19 15:31**

