



**embarcadero®**

**Interbase 2020**

**Sincronização de Dados Simples, Rápida e Segura!**

Fernando Rizzato  
Lead Software Consultant, LatAm

Sriram Balasubramanian  
Architect/Engineering Team Lead, InterBase



For quality purposes, all lines except the presenter are muted



IT'S OK TO ASK QUESTIONS! Use the Q&A Panel on the Right

*This webinar is being recorded for future playback.  
Recordings will be available on Embarcadero's YouTube channel*

# AGENDA



InterBase Hoje



InterBase para  
*Developers*



Recursos em  
Destaque



Novas  
Plataformas

InterBase é um banco de dados SQL ultrarrápido, escalável e “incorporável”, com recursos de segurança de dados de nível corporativo, recuperação de desastres e sincronização de mudanças. Está disponível como um produto autônomo e em nossas ferramentas: RAD Studio, Delphi e C++Builder

# INTERBASE® 2020



32bit &  
64bit  
Intel



32bit &  
64bit  
Intel



32bit &  
64bit  
Intel



32bit &  
64bit  
Intel



32bit &  
64bit  
Intel



32bit &  
64bit  
Intel



32bit &  
64bit  
ARM



32bit &  
64bit  
ARM



Server

InterBase ToGo / IBLite

# InterBase 2017 e posteriores



## Administração quase zero

- ✓ Alta escalabilidade, poderoso, com opção de uso embutido
- ✓ Baixo consumo de memória e dados em disco
- ✓ Instalação e distribuição simples
- ✓ Mecanismo de criptografia em todas as edições comerciais
- ✓ Administração simples
- ✓ Inteligência de auto ajuste
- ✓ Baixo TCO

## Segurança/Criptografia

- ✓ Usuário & autenticação baseada em regras
- ✓ Criptografia total, em tabelas e/ou colunas
  - DES (weak)
  - AES (strong 256bit) encryption
- ✓ Descriptografia pelo user / user role
- ✓ Permite ao desenvolvedor manter foco na criação rápida de aplicações.

## Recuperação de Desastre

- ✓ Journaling & Point In time recovery
- ✓ Data dumps
- ✓ Incremental data dumps

# InterBase 2017 e posteriores



## Integração e conectividade

- ✓ Ferramentas de Desenvolvimento Embarcadero
- ✓ Ferramentas de Banco de Dados Embarcadero
- ✓ Visual Studio, Eclipse, etc.
- ✓ ODBC, JDBC
- ✓ 100% compatível com o padrão SQL
- ✓ 100% ACID compatível

## Auto-ajuste e performance

- ✓ Multi-version architecture
- ✓ Recuperação automática de falhas
- ✓ SMP - Multiple core suporte
- ✓ Monitoramento de performance
- ✓ Consultas para métrica de performance de dados

## Change Views

- ✓ Poderoso acompanhamento de mudanças
- ✓ Assinatura individual de uma fotografia
- ✓ Rastreamento de mudança de dados em uma tabela e/ou uma coluna
- ✓ Grant subscription privilege to users/roles

# O QUE HÁ DE NOVO NO IB 2020?

- Database Tablespaces
- Monitoramento de performance/métrica de utilização de index
- Dicionário de dados DDL
- Otimização SQL
- Melhorias de Segurança- OpenSSL atualizado
- Melhorias de suporte no IBConsole
- Novas plataformas suportadas
  - Android 64-bit
  - macOS 64-bit



# Database Tablespaces

# *Sample Schema- Database with no Tablespaces*

```
/* No defined Tablespaces */
CREATE TABLE T1 (F1 INTEGER DEFAULT NULL, F2 INTEGER DEFAULT NULL, F3 VARCHAR(20) DEFAULT NULL);
CREATE INDEX IDX_T1 ON T1(F1);
COMMIT;

CREATE TABLE T2 (F1 INTEGER DEFAULT NULL, F2 INTEGER DEFAULT NULL, F3 VARCHAR(20) DEFAULT NULL);
CREATE INDEX IDX_T2 ON T2(F1);
COMMIT;

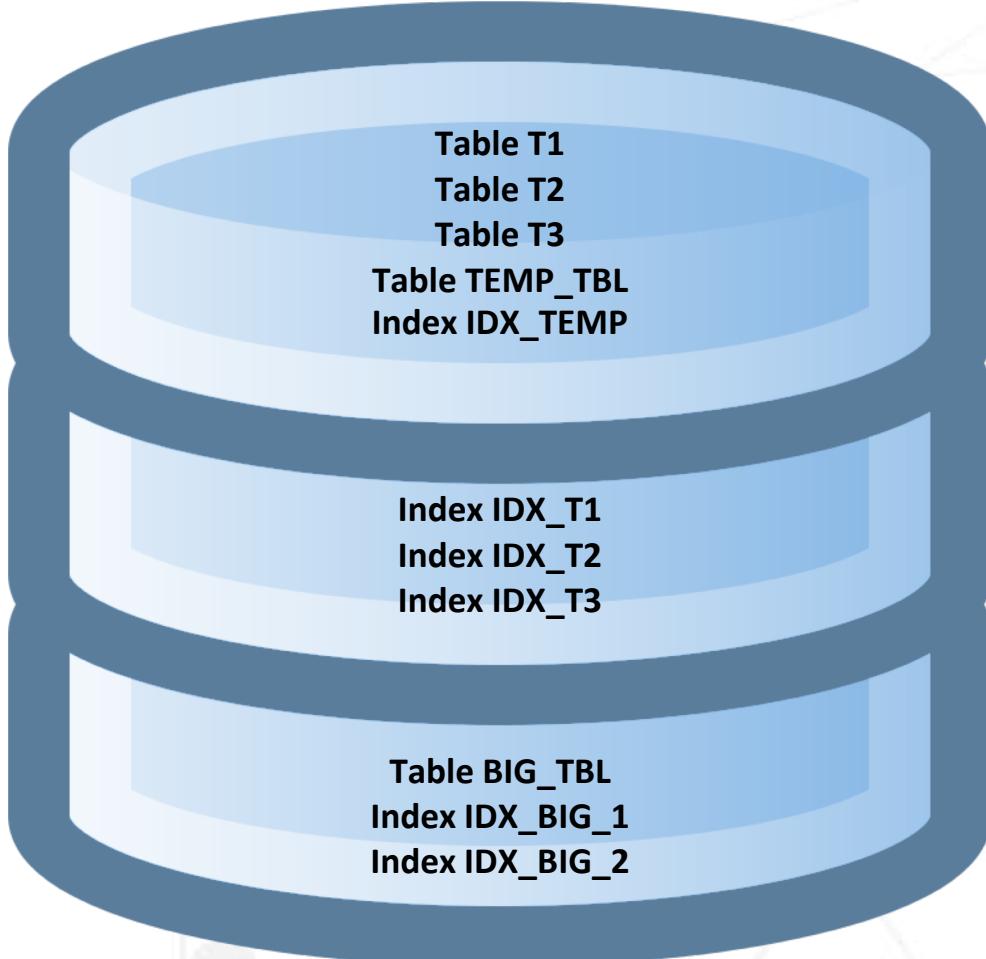
CREATE TABLE T3 (F1 INTEGER DEFAULT NULL, F2 INTEGER DEFAULT NULL, F3 VARCHAR(20) DEFAULT NULL);
CREATE INDEX IDX_T3 ON T3(F1);
COMMIT;

CREATE TABLE BIG_TBL (F1 INTEGER DEFAULT NULL, F2 INTEGER DEFAULT NULL, F3 VARCHAR(20000) DEFAULT NULL);
CREATE INDEX IDX_BIG_1 ON BIG_TBL(F1);
CREATE INDEX IDX_BIG_2 ON BIG_TBL(F2);
COMMIT;

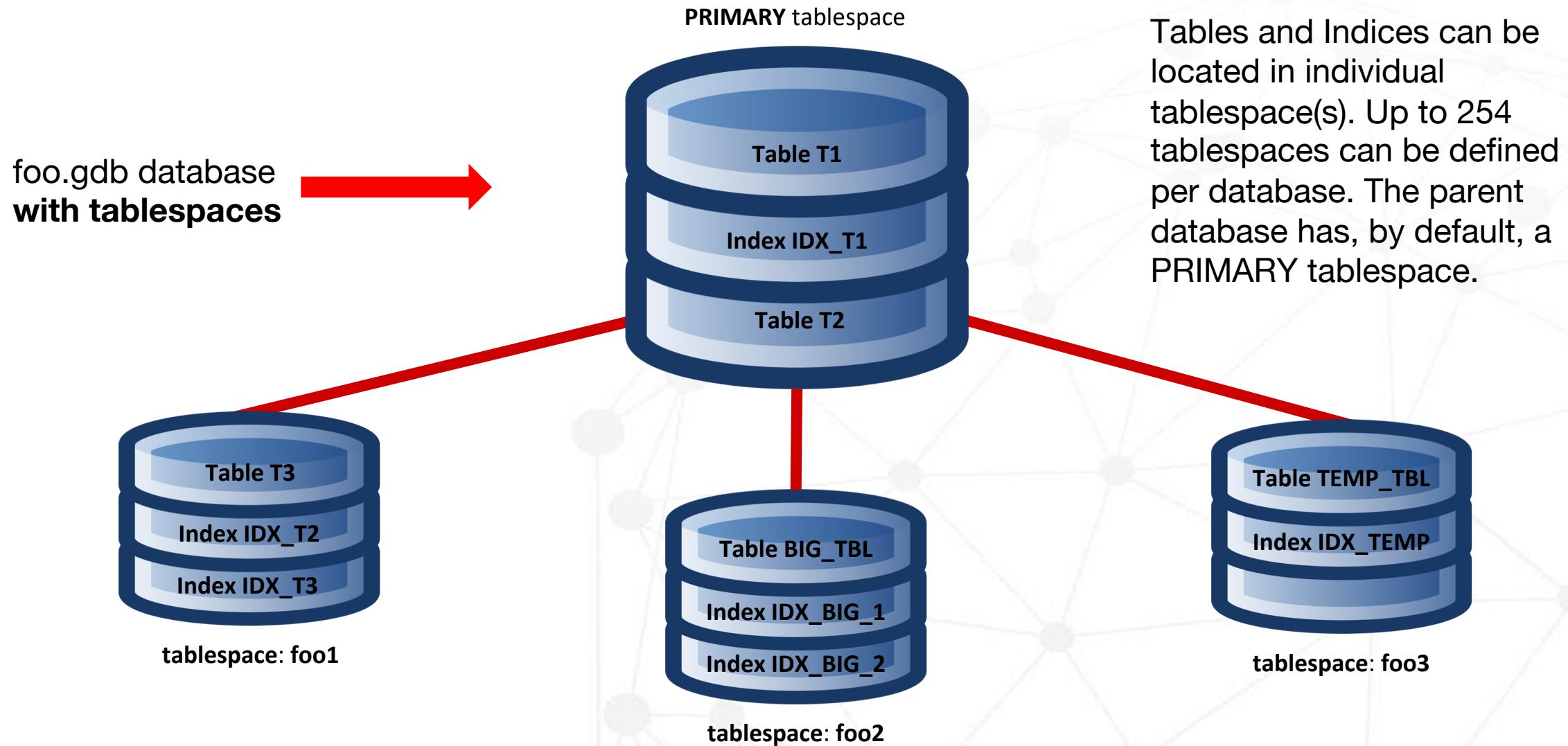
CREATE GLOBAL TEMPORARY TABLE TEMP_TBL (F1 INTEGER DEFAULT NULL, F2 INTEGER DEFAULT NULL) ON COMMIT PRESERVE
ROWS NO RESERVE SPACE;
CREATE INDEX IDX_TEMP ON TEMP_TBL(F1);
COMMIT;
```

# *Database Tablespace*

foo.gdb single file  
database **without**  
**tablespaces**



# Database Tablespace



# *Sample - Enabling Tablespaces*

```
/* Tablespaces */
CREATE TABLESPACE foo1 FILE 'foo1' NO RESERVE SPACE;
CREATE TABLESPACE foo2 FILE 'foo2';
CREATE TABLESPACE foo3 FILE 'foo3';
COMMIT;

/* Move tables and indices to their target tablespace destinations */
ALTER TABLE T1 ALTER TABLESPACE PRIMARY;
ALTER INDEX IDX_T1 ALTER TABLESPACE PRIMARY;
COMMIT;

ALTER TABLE T2 ALTER TABLESPACE PRIMARY;
ALTER INDEX IDX_T2 ALTER TABLESPACE foo1;
COMMIT;

ALTER TABLE T3 ALTER TABLESPACE foo1;
ALTER INDEX IDX_T3 ALTER TABLESPACE foo1;
COMMIT;

ALTER TABLE BIG_TBL ALTER TABLESPACE foo2;
ALTER INDEX IDX_BIG_1 ALTER TABLESPACE foo2;
ALTER INDEX IDX_BIG_2 ALTER TABLESPACE foo2;
COMMIT;

ALTER TABLE TEMP_TBL ALTER TABLESPACE foo3;
ALTER INDEX IDX_TEMP ALTER TABLESPACE foo3;
COMMIT;
```

# *Sample - Show Tablespace info in ISQL*

```
SHOW TABLESPACE;
SHOW TABLESPACES;
SHOW TABLESPACE primary;
SHOW TABLESPACE foo1;
SHOW TABLESPACE foo2;
SHOW TABLESPACE foo3;

SHOW TABLES IN TABLESPACE;
SHOW TABLES IN TABLESPACE primary;
SHOW TABLES IN TABLESPACE foo1;
SHOW TABLES IN TABLESPACE foo2;
SHOW TABLES IN TABLESPACE foo3;

SHOW INDICES IN TABLESPACE;
SHOW INDICES IN TABLESPACE primary;
SHOW INDICES IN TABLESPACE foo1;
SHOW INDICES IN TABLESPACE foo2;
SHOW INDICES IN TABLESPACE foo3;
```

## *Tablespace Backup sample*

Tablespace backup files are created when the *tablespace* switch is used.

```
/* Backup HR and EMPLOYEE tablespaces to single file */
```

```
gbak -b employee.idb hr_eh.itbk +tablespace HR +tablespace  
EMPLOYEE
```

# *Tablespace Backup/Restore samples*

```
/* Restore an individual tablespace by truncating its  
tables in place without deleting the existing  
tablespace file. */
```

```
gbak -create_tablespace employee.idbk employee.idb +tablespace  
EMPLOYEE
```

```
/* Restore an individual tablespace by deleting the  
existing tablespace file and creating a new file into  
which its tables are restored. */
```

```
gbak -replace_tablespace employee.idbk employee.idb +tablespace  
EMPLOYEE
```

“

## *Index Usage Metrics Via Performance Monitoring*



# *Index Usage Metrics Via Performance Monitoring*

**TMP\$INDICES** is a new system table included as part of the Performance Monitoring system tables. This table tracks all loaded indices per table in the database with key metrics for monitoring.

```
/* List all indices for a specific relation/table */
SELECT * FROM TMP$INDICES
WHERE TMP$RELATION_NAME='foo';

/* List all indices with depth greater than 3 */
SELECT * FROM TMP$INDICES
WHERE TMP$INDEX_DEPTH > 3;

/* List all indices with more than 10 segments */
SELECT * FROM TMP$INDICES
WHERE TMP$INDEX_SEGMENTS > 10;

/* List all indices that are index type 'FOREIGN KEY' */
SELECT * FROM TMP$INDICES
WHERE TMP$INDEX_TYPE='FOREIGN KEY';

/* List all indices with tons of delete operations leading to reverse page splits.
This is a good indicator that index selectivity needs to be recalculated */
SELECT * FROM TMP$INDICES
WHERE TMP$PAGE_REVERSE_SPLITS > 1000;
```

“

## *Data Dictionary DDL*



# Data Dictionary DDL

## ALTER DESCRIPTION FOR samples

```
/* Use ALTER DESCRIPTION command with description set to text */
alter description for database set 'Database description sample';
alter description for exception customer_check set 'Exception msg: Checking
on customer sample';
alter description for filter desc_filter set 'Filter description sample';
alter description for index custnamex set 'Index description sample';
alter description for procedure add_emp_proj set 'Stored Procedure
description sample';
alter description for table employee set 'Table description sample';
alter description for table "MyTableDelim" set 'Table delimited identifier
description sample';
alter description for trigger set_cust_no set 'Trigger description sample';
alter description for external function abs set 'UDF description sample';
alter description for user sysdso set 'User description sample';
alter description for subscription sub_ceo_multidevice set 'Subscription
description sample';
alter description for encryption backup_key set 'Encryption key description
sample';
```

“

## *SQL Optimizations*



# *SQL Optimizations*

Transform inequality operators and NOT operators to their opposite representation for index based retrieval. Inequality operators, such as <, <=, !=, <>, >, >= can be transformed and optimized.

```
select * from tone where NOT F1>=2;  
select * from tone where NOT F1>2 and F2>0;  
select * from tone where NOT F1>=2 and F2>0;  
select * from tone where NOT F1<3;  
select * from tone where NOT F1<=3;  
select * from tone where NOT F1<2;  
select * from tone where NOT F1<2 ORDER BY F1;  
select * from tone where NOT F1<2 ORDER BY F1 DESC;  
select * from tone where NOT F1>2 ORDER BY F1;  
select * from tone where NOT F1>2 ORDER BY F1 DESC;  
select * from tone where NOT F1>=2 ORDER BY F1;  
select * from tone where NOT F1>=2 ORDER BY F1 DESC;  
select * from tone where NOT F1<2 AND F2<=2;  
select * from tone where NOT F1<=2 AND F2<=2;  
select * from tone where NOT F1<2 AND NOT F2<=2;  
select * from tone where NOT F1<=2 AND NOT F2<=2;  
select * from tone where NOT F1<=3 OR NOT F1>=2;  
select * from tone where NOT F1 BETWEEN 2 and 3;  
select * from tone where NOT F1 IN (1,4);  
select * from tone where NOT NOT F1<=3;
```

“

## *Melhorias no IBConsole*



# *Melhorias no IBConsole*

- InterBase 2020 features support (Tablespaces, Alter Description DDL)
- Change Views subscription context and coloring of change status in WISQL
- 64-bit IBConsole.exe binary with 64-bit Edition of InterBase
- Syntax highlighting of queries
- Server-wide Performance Monitoring enhancements
- Closing the Interactive SQL Window on a read only transaction no longer prompts a commit or rollback dialog, transaction is committed and the window closes.
- Enhanced support for the RECONNECT clause, it now supports CACHE, lc\_ctype and DIALECT.
- A new Show tab - this tab shows information from WISQL SHOW statements

CHANGE VIEWS DEMO

“

*Obrigado!!!*

[fernando.rizzato@embarcadero.com](mailto:fernando.rizzato@embarcadero.com)

