

SQL VČERA A DNES ...

Petr Jiroušek
(petr@civ.zcu.cz)

O čem to vlastně de řeč

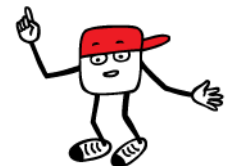
- SQL = Structured Query Language
- Jazyk pro komunikaci uživatelů (rozuměj spíše aplikací) s databázemi
- Vychází z běžné hovorové angličtiny, je tudíž snadno srozumitelný
- Je dnes v této oblasti světovým standardem č. 1



SQL včera (SQL 89)

SQL 89 mělo cca 100 reserved words:

ABS | ALL | AND | ANY | AS | ASC | AUTHORIZATION | AVG
BEGIN | BETWEEN | BY
CHAR | CHARACTER | CHECK | CLOSE | COBOL | COMMIT | CONTINUE | COUNT | CREATE |
CURRENT | CURSOR
DEC | DECIMAL | DECLARE | DEFAULT | DELETE | DESC | DISTINCT | DOUBLE
END | ESCAPE | EXEC | EXISTS
FETCH | FLOAT | FOR | FOREIGN | FORTRAN | FOUND | FROM
GO | GOTO | GRANT | GROUP
HAVING
IN | INDICATOR | INSERT | INT | INTEGER | INTO | IS
KEY
LANGUAGE | LIKE
MAX | MIN | MODULE |
NOT | NULL | NUMERIC
OF | ON | OPEN | OPTION | OR | ORDER
PASCAL | PLI | PRECISION | PRIMARY | PRIVILEGES | PROCEDURE | PUBLIC
REAL | REFERENCES | ROLLBACK
SCHEMA | SECTION | SELECT | SET | SMALLINT | SOME | SQL | SQLCODE | SQLERROR | SUM
TABLE | TO
UNION | UNIQUE | UPDATE | USER
VALUES | VIEW
WHENEVER | WHERE | WITH | WORK

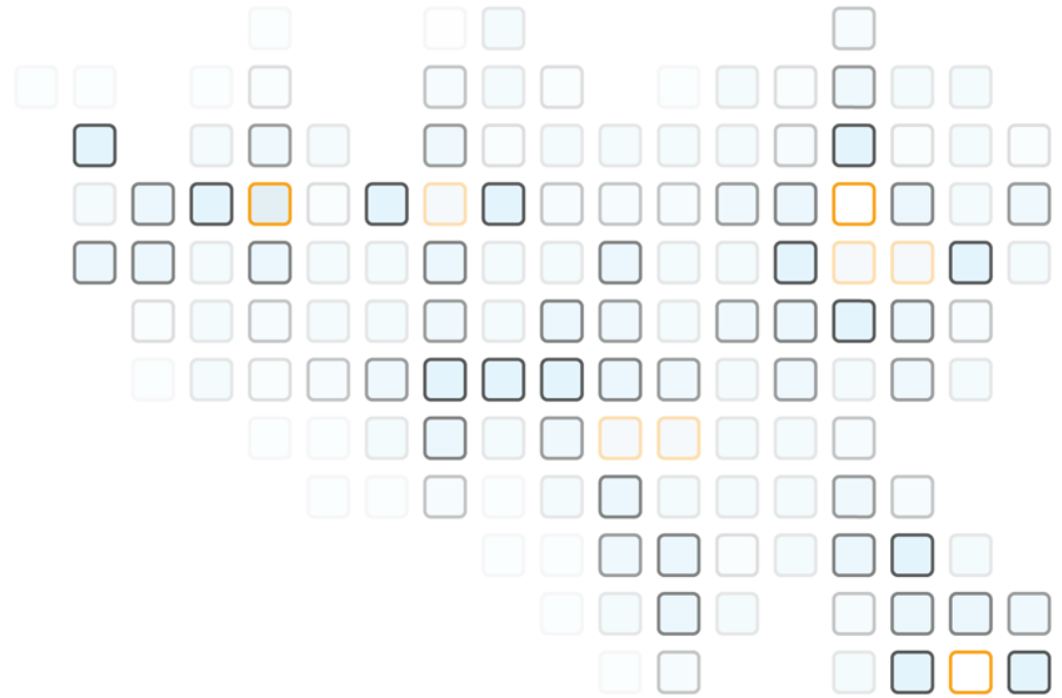


SQL dnes (2011)

SQL 2011 má cca 350 reserved words:


|ABS|ALL|ALLOCATE|ALTER|AND|ANY|ARE|ARRAY|ARRAY_AGG|ARRAY_MAX_CARDINALITY|AS|ASENSITIVE|ASYMMETRIC|AT|ATOMIC
|AUTHORIZATION| AVG
|BEGIN|BEGIN_FRAME|BEGIN_PARTITION|BETWEEN|BIGINT|BINARY| BLOB|BOOLEAN|BOTH|BY
|CALL|CALLED|CARDINALITY|CASCADED|CASE|CAST|CEIL|CEILING| CHAR|CHAR_LENGTH|CHARACTER|CHARACTER_LENGTH|CHECK|CLOB|CLOSE
|COALESCE|COLLATE|COLLECT|COLUMN|COMMIT|CONDITION|CONNECT|CONSTRAINT|CONTAINS|CONVERT|CORR|CORRESPONDING|COUNT
|COVAR_POP| COVAR_SAMP|CREATE|CROSS|CUBE|CUME_DIST|CURRENT|CURRENT_CATALOG|CURRENT_DATE
|CURRENT_DEFAULT_TRANSFORM_GROUP|CURRENT_PATH | CURRENT_ROLE| CURRENT_ROW|CURRENT_SCHEMA|CURRENT_TIME
|CURRENT_TIMESTAMP| CURRENT_TRANSFORM_GROUP_FOR_TYPE|CURRENT_USER|CURSOR|CYCLE
|DATE|DAY|DEALLOCATE|DEC|DECIMAL|DECLARE|DEFAULT|DELETE| DENSE_RANK|DEREF|DESCRIBE|DETERMINISTIC|DISCONNECT|DISTINCT
|DOUBLE|DROP | DYNAMIC
|EACH|ELEMENT|ELSE|END|END_FRAME|END_PARTITION|END-EXEC| EQUALS|ESCAPE|EVERY|EXCEPT|EXEC|EXECUTE|EXISTS|EXP| EXTERNAL
|EXTRACT
|FALSE|FETCH|FILTER|FIRST_VALUE|FLOAT|FLOOR|FOR|FOREIGN| FRAME_ROW|FREE|FROM|FULL|FUNCTION|FUSION
|GET|GLOBAL|GRANT|GROUP|GROUPING|GROUPS
|HAVING|HOLD|HOUR| IDENTITY|IN|INDICATOR|INNER|INOUT|INSENSITIVE|INSERT
|INT|INTEGER|INTERSECT|INTERSECTION|INTERVAL|INTO|IS
|JOIN
|LAG|LANGUAGE|LARGE|LAST_VALUE|LATERAL|LEAD|LEADING|LEFT| LIKE|LIKE_REGEX|LN|LOCAL|LOCALTIME|LOCALTIMESTAMP|LOWER
|MATCH|MAX|MEMBER|MERGE|METHODO|MIN|MINUTE|MOD|MODIFIES|MODULE|MONTH|MULTISET
|NATIONAL|NATURAL|NCHAR|NCLOB|NEW|NO|NONE|NORMALIZE|NOT| NTH_VALUE|NTILE|NULL|NULLIF|NUMERIC
|OCTET_LENGTH|OCCURRENCES_REGEX|OF|OFFSET|OLD|ON|ONLY|OPEN| OR|ORDER|OUT|OUTER|OVER|OVERLAPS|OVERLAY
|PARAMETER|PARTITION|PERCENT|PERCENT_RANK|PERCENTILE_CONT|PERCENTILE_DISC|PERIOD|PORTION|POSITION|POSITION_REGEX
|POWER|PRECEDES| PRECISION| PREPARE|PRIMARY|PROCEDURE
|RANGE|RANK|READS|REAL|RECURSIVE|REF|REFERENCES|REFERENCING|REGR_AVGX|REGR_AVGY|REGR_COUNT|REGR_INTERCEPT|REGR_R2
|REGR_SLOPE|REGR_SXX|REGR_SXY|REGR_SYY|RELEASE|RESULT|RETURN|RETURNS| EVOKE|RIGHT|ROLLBACK|ROLLUP|ROW|ROW_NUMBER|ROWS
|SAVEPOINT|SCOPE|SCROLL|SEARCH|SECOND|SELECT| SENSITIVE|SESSION_USER|SET|SIMILAR|SMALLINT|SOME|SPECIFIC| SPECIFICTYPE|SQL
|SQLEXCEPTION|SQLSTATE|SQLWARNING|SQRT|START| STATIC|STDDEV_POP|STDDEV_SAMP|SUBMULTISET|SUBSTRING|SUBSTRING_REGEX
|SUCCEEDS|SUM|SYMMETRIC|SYSTEM|SYSTEM_TIME|SYSTEM_USER
|TABLE|TABLESAMPLE|THEN|TIME|TIMESTAMP|TIMEZONE_HOUR|TIMEZONE_MINUTE|TO|TRAILING|TRANSLATE|TRANSLATE_REGEX|TRANSLATION
|TREAT| TRIGGER|TRUNCATE|TRIM|TRIM_ARRAY|TRUE
|UESCAPE|UNION|UNIQUE|UNKNOWN|UNNEST|UPDATE|UPPER|USER|USING
|VALUE|VALUES|VALUE_OF|VAR_POP|VAR_SAMP|VARBINARY| VARCHAR|VARYING|VERSIONING
|WHEN|WHenever|WHERE|WIDTH_BUCKET|WINDOW|WITH|WITHIN|WITHOUT
|YEAR





NO A TO JE V KOSTCE VŠE

Co je to databáze

- Moře informací ...
 - Co je to informace ?
 - Účel databází ?
- To jsem teda zvědavá ...
- 
- A cartoon illustration of a woman with short brown hair, wearing a brown top and a red skirt. She has a questioning expression on her face and is holding a small black object to her ear. A speech bubble above her contains the text "To jsem teda zvědavá ...".
- Vezmeme to od začátku ... ten bude pro všechny, pak se to maličko zkomplikuje a konec bude již jen pro opravdové databázové odborníky ...

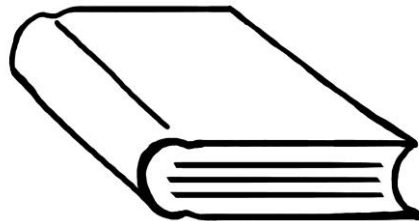
Pravěk

- Lidé žili v tlupách, jedli syrové maso, spali v jeskyních
- Měli oheň, jinak by umřeli zimou a hladu
- O oheň se musel pořád někdo starat, kdyby vyhasl, znamenalo by to konec kmene



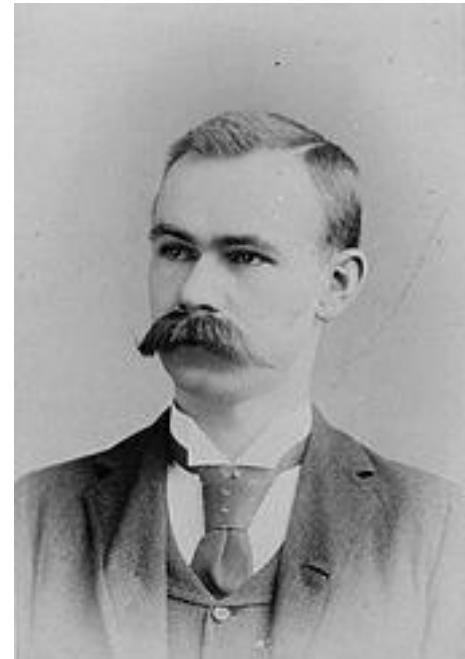
Informace

- Informace o tom, že oheň se dá nejen získat při požáru, ale i vyrobit, se předávala dál
- Časem to někdo vytesal do kamene, napsal do knih, uložil na wikipedii ...
- Postupně těch informací bylo ale tolik, že bylo potřeba vybudovat mechanismus pro ukládání jednotlivých informací a získání té správné informace co možná nejrychleji



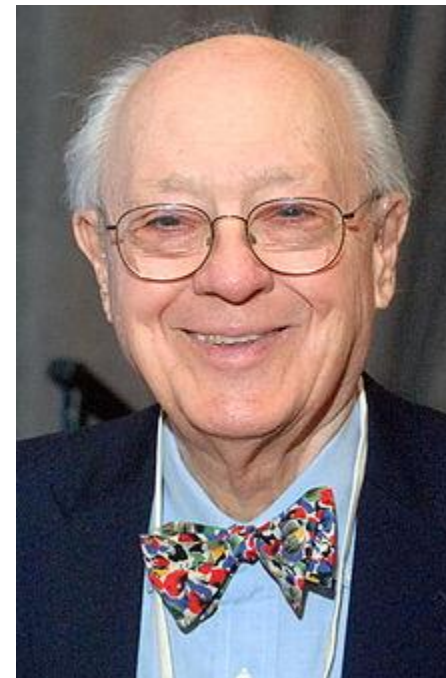
Opravdové databáze

- Rokem **R** je zřejmě rok 1890
- **Herman Hollerith** v tomto roce poprvé použil elektromechanický stroj ke sčítání lidu v USA
- Tyto stroje měly automatizovaný zápis a možnost vyhledávání
- Paměťovým médiem bylo cosi jako dřevný štítek



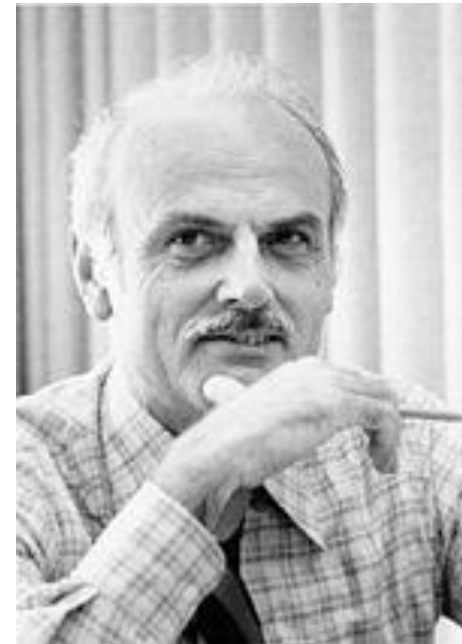
Nějaká další jména

- Každá přednáška o historii by měla obsahovat nějaká jména slavných
- Už jsem zmiňoval „otce“ databází Hermana Holleritha
- **Charles Bachman** z General Electric v roce 1961 představil svůj první datový sklad
- Na počátku šedesátých let byl Bachman též u založení seskupení Codasyl, které v roce 1965 publikovalo specifikaci jazyka COBOL



Nějaká další jména

- Další důležitým mužem byl **Ted Codd** z IBM
- Na počátku sedmdesátých let představil svůj koncept neprocedurálního jazyka pro přístup k datům, který již obsahuje dnes běžně používané rozdělení příkazů na DDL (data definition language) pro vytváření databázových objektů a DML (data manipulation language) pro manipulaci s daty (select, update, insert, delete)



Nějaká další jména

- A poslední dvě jména:
 - **Donald D. Chamberlin**
 - **Raymond F. Boyce**
- V roce 1974 představili svůj jazyk SEQUEL
- A od něj již byl je krůček k tomu, co dnes známe jako SQL



Stručná historie SQL

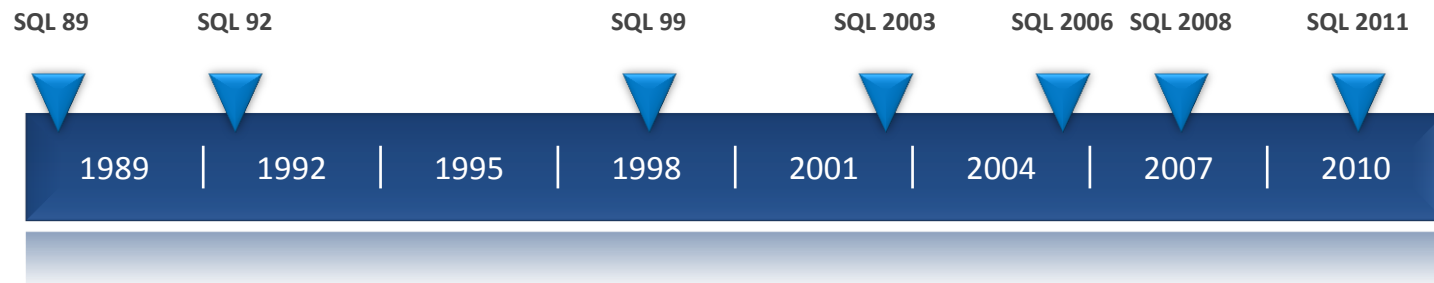
- **První** implementaci jazyka SQL představila firma RSI (Relational Software Inc.) v roce 1979
 - OracleV2 (Version2) pro počítače VAX
- Teprve následně uvedlo svou vlastní implementaci SQL na trh i IBM
 - Jednotlivé verze byly pojmenovány System/38, SQL/DS a DB2 (v letech 1979, 1981 a 1983)

Že by IBM takhle zaspalo ?
To snad ani není možné ...



Stručná historie SQL

- IBM chtělo být alespoň v něčem první a když už se jí to nepovedlo s databází ...
- V roce 1986 byl v režii IBM první pokus o standardizaci jazyka SQL
- V roce 1989 vzniká již pod hlavičkou ANSI první všeobecně uznávaný standard
- Další ANSI/ISO standardy následují v letech 1992, 1999, 2003, 2006, 2008, 2011



Jednotlivé části SQL standardu

- ISO/IEC 9075-1:2011 Part 1: *Framework* (SQL/Framework) – 90 stránek
 - Definuje základní pojmy jako SQL server, SQL klient, SQL schéma, catalog, druhy datových typů – číselné, znakové, binární, logické ..., null hodnota, SQL objekty – tabulka, pohled, trigger, doména, SQL routines, sekvence ..., práva, constraint
- ISO/IEC 9075-2:2011 Part 2: *Foundation* (SQL/Foundation) – 1483 stránek
 - Vlastní konstrukce SQL jazyka
 - Nejobsáhlejší a vlastně základní část
 - Vazby na jazyky: Ada, C, COBOL, Fortran, MUMPS, Pascal, and PL/I
- ISO/IEC 9075-3:2008 Part 3: *Call-Level Interface* ([SQL/CLI](#)) – 409 stránek
 - Příkazy CLI – otvírání kurzorů, parsování, zpracování dat, návratové kódy ..., diagnostika, v příloze vazby na jazyky C a COBOL (java je v části 10)
 - Tato část je podmnožinou ODBC standardu
- ISO/IEC 9075-4:2011 Part 4: *Persistent Stored Modules* ([SQL/PSM](#)) – 200 stránek
 - Vychází z části 2 a upřesňuje, za jakých podmínek a s jakými dalšími parametry lze dané SQL příkazy použít ve Stored Modules, doplňuje konstrukce pro definici proměnných, porovnávání, smyčky ...

Jednotlivé části SQL standardu

- ISO/IEC 9075-9:2008 Part 9: *Management of External Data* ([SQL/MED](#)) – 491 stránek
 - Definuje pojmy foreign server, datalink, foreign server request ...
 - Je postaven nad částmi 2 a 3, definuje zda a za jakých podmínek lze klasické SQL příkazy vykonávat vzdáleně a rozšiřuje CLI o příkazy pro vzdálené zpracování dat
- ISO/IEC 9075-10:2008 Part 10: *Object Language Bindings* ([SQL/OLB](#)) – 417 stránek
 - Definuje vazby na SQL na javu tzv. SQLJ = SQL/OLB
 - JDBC je jiný standard!
- ISO/IEC 9075-11:2011 Part 11: *Information and Definition Schemas* ([SQL/Schemata](#)) – 316 stránek
 - DEFINITION_SCHEMA – obsahuje datový model, který je v dané DB obsažen
 - INFORMATION_SCHEMA – zajišťuje uživatelský přístup (na bázi SQL view) přístup k DEFINITION_SCHEMA

Jednotlivé části SQL standardu

- ISO/IEC 9075-13:2008 Part 13: *SQL Routines and Types Using the Java Programming Language* ([SQL/JRT](#)) – 210 stránek
 - Definice použití jazyka java přímo v databázích
- ISO/IEC 9075-14:2011 Part 14: *XML-Related Specifications* ([SQL/XML](#)) – 463 stránek
 - XML datové typy (včetně schématu a DTD), XML funkce, Xquery, XML mapování dat (export dat i metadat v XML), XML DEFINITION_SCHEMA a INFORMATION_SCHEMA, vazby na jazyky Ada, C, COBOL, Fortran, Pascal, PL/I

Nevypadl náhodou jeden slajd?

- Kdo pozorně sleduje přednášku, patrně si všimnul, že po části 4 následovala část 8
- Není to chyba, ten slajd opravdu nevypadl 😊
- Part 5 - SQL/Bindings – byl publikován v SQL 99, ale v roce 2003 byl sloučen do Part 2: *Foundation*
- Part 6 - SQL/Transactions – měl specifikovat distribuované transakce, práce byly započaty, ale projekt odumřel pro nezáměr, část je v Part 9: *Management of External Data*
- Part 7 - SQL/Temporal – proces standardizace začal v roce 1995, ale v roce 2001 pro nezáměr skončil, Temporal Data se dostaly v roce 2011 přímo do Part 2: *Foundation*
- Part 8 - SQL/Objects Extended Objects prakticky okamžitě byl přesunut do Part 2: *Foundation*

SQL Committee (2011)

- **Officers**
 - Chair: Don Deutsch – Oracle Corporation
 - Vice Chair: Keith Hare – JCC Consulting, Inc.
 - Secretary – Mike Gorman – Whitemarsh
 - International Representative – Krishna Kulkarni – IBM
 - Editor – Jim Melton – Oracle Corporation
- **Participating Companies**
 - Actian
 - Bentley Systems
 - CA
 - Greenplum
 - HP
 - IBM
 - Intersystems
 - JCC Consulting, Inc.
 - Microsoft
 - Oracle Corporation
 - Software Workshop, Inc.
 - Sybase
 - Teradata
 - Whitemarsh Information Systems

SQL 86

- Bylo v režii IBM (rozuměj šito na míru DB2)
- Definuje způsob zpracování DML
 - *Nedefinuje DDL*
- Tento standard byl hodně kritizován právě pro silnou vazbu na IBM
- Institut ANSI/ISO jej adoptoval až cca rok po jeho vydání IBM

Pročpak až po roce?



SQL 89

- SQL-89, Standard SQL, SQL-1 je první relativně všeobecně uznávaný SQL standard
- Definuje DDL (create table, index ...)
 - *Ještě nemá DROP nebo ALTER*
- Přidává práva GRANT/REVOKE
- Umožňuje jednoduché řízení transakcí (begin, commit, rollback)
- Zavádí pojem kurzor
- Definuje referenční integritu (unikátní klíče, cizí klíče a check constrainty)
- Vazby na jazyky COBOL, Fortan, Pascal, PL/1

ANSI 92

- Standard z roku 1992 je znám pod označením ANSI SQL 92
- Je stále ve formě jedné normy – 685 stran
- Na jednotlivé části se rozdělí až postupně v dalších letech

Teď jen zjistit jak se to dá
hacknout ...



ANSI 92

- Datové typy: DATE, TIME, TIMESTAMP, INTERVAL, BIT string, VARCHAR strings, a NATIONAL CHARACTER strings.
- Podpora znakových sad
- Nové funkce např. konkatenace řetězců, substring, funkce pro práci s typy DATE a TIME
- Příkaz CASE
- Nové typy spojování tabulek v selectu (UNION JOIN, NATURAL JOIN)
- Příkazy ALTER a DROP
- Vazba na další jazyky: C, Ada a MUMPS.
- INFORMATION_SCHEMA – obsahuje metadata o všech objektech v DB
- Dynamické dotazy
- CREATE TEMP TABLE

ANSI 92

SQL 92 mělo cca 230 reserved words

|ABSOLUTE|ACTION|ADD|ALL|ALLOCATE|ALTER|AND|ANY|ARE|AS|ASC|ASSERTION|AT|AUTHORIZATION|AVG
|BEGIN|BETWEEN|BIT|BIT_LENGTH|BOTH|BY
|CASCADE|CASCADED|CASE|CAST|CATALOG|CHAR|CHARACTER|CHAR_LENGTH|CHARACTER_LENGTH|CHECK|CLOSE|COALESCE|COLLATE
|COLLATION|COLUMN|COMMIT|CONNECT|CONNECTION|CONSTRAINT|CONSTRAINTS|CONTINUE|CONVERT|CORRESPONDING|COUNT|CREATE
|CROSS|CURRENT|CURRENT_DATE|CURRENT_TIME|CURRENT_TIMESTAMP|CURRENT_USER|CURSOR
|DATE|DAY|DEALLOCATE|DEC|DECIMAL|DECLARE|DEFAULT|DEFERRABLE|DEFERRED|DELETE|DESC|DESCRIBE|DESCRIPTOR|DIAGNOSTICS
|DISCONNECT|DISTINCT|DOMAIN|DOUBLE|DROP
|ELSE|END|END-EXEC|ESCAPE|EXCEPT|EXCEPTION|EXEC|EXECUTE|EXISTS|EXTERNAL|EXTRACT
|FALSE|FETCH|FIRST|FLOAT|FOR|FOREIGN|FOUND|FROM|FULL
|GET|GLOBAL|GO|GOTO|GRANT|GROUP
|HAVING|HOUR
|IDENTITY|IMMEDIATE|IN|INDICATOR|INITIALLY|INNER|INPUT|INSENSITIVE|INSERT|INT|INTEGER|INTERSECT|INTERVAL|INTO|IS|ISOLATION
|JOIN
|KEY
|LANGUAGE|LAST|LEADING|LEFT|LEVEL|LIKE|LOCAL|LOWER
|MATCH|MAX|MIN|MINUTE|MODULE|MONTH
|NAMES|NATIONAL|NATURAL|NCHAR|NEXT|NO|NOT|NULL|NULLIF|NUMERIC
|OCTET_LENGTH|OF|ON|ONLY|OPEN|OPTION|OR|ORDER|OUTER|OUTPUT|OVERLAPS
|PAD|PARTIAL|POSITION|PRECISION|PREPARE|PRESERVE|PRIMARY|PRIOR|PRIVILEGES|PROCEDURE|PUBLIC
|READ|REAL|REFERENCES|RELATIVE|RESTRICT|REVOKE|RIGHT|ROLLBACK|ROWS
|SCHEMA|SCROLL|SECOND|SECTION|SELECT|SESSION|SESSION_USER|SET|SIZE|SMALLINT|SOME|SPACE|SQL|SQLCODE|SQLERROR
|SQLSTATE|SUBSTRING|SUM|SYSTEM_USER
|TABLE|TEMPORARY|THEN|TIME|TIMESTAMP|TIMEZONE_HOUR|TIMEZONE_MINUTE|TO|TRAILING|TRANSACTION|TRANSLATE|TRANSLATION
|TRIM|TRUE
|UNION|UNIQUE|UNKNOWN|UPDATE|UPPER|USAGE|USER|USING
|VALUE|VALUES|VARCHAR|VARYING|VIEW
|WHEN|WHENEVER|WHERE|WITH|WORK|WRITE
|YEAR
|ZONERDER

To se přeci nikdo nemůže naučit!!!



Rozšiřování ANSI 92

- V roce 1995 je norma ANSI 92 rozšířena o SQL/CLI (Call Level Interface)
 - později Part 3
- V roce 1996 je norma ANSI 92 rozšířena o SQL/PSM (stored procedures)
 - později Part 4

A pak konečně přišla větší revize 99

- V devadesátých letech nastal boom databází
- Databáze vyráběl „kdo měl ruce a nohy“
- Možná znáte ... Oracle, IBM, Informix, Postgres, Ingres, Berkeley DB ... vzniká silný konkurenční boj
- Na pořad dne se dostává nutnost kompatibility jednotlivých databází na „lepší úrovni“ než jen používání stejných příkazů
- Na standardizaci se podílejí všichni „velcí hráči na trhu“
- Standard významně nabobtnává a je rozdělen na jednotlivé části

SQL 99

- Nové datové typy: Time-Series, User defined. LOB, CLOB, BLOB
- Role
- CASCADE DELETE
- DISTINCT
- SQLJ for Java
- Triggery
- CLI kompatibilita se standardy ODBC, JDBC, OLE DB
- Stored Procedures
- Datalinky (propojování databází)
- GROUP BY ROLLUP, CUBE
- Savepoint

SQL 2003

- Nastává doba XML, XML je všude



SQL 2003

- Obsahuje hodně malých změn a jednu velkou – XML (292 stránek)
 - obsahuje elementy a struktury pro popis tabulky, indexu, volání procedury ... prostě v XML lze popsat vše, co se v DB nachází
 - Obsahuje i XML funkce, ale v tomto standardu jich je ještě relativně málo – brzy se ukáže, že je to špatně
- Dále např. CREATE TABLE AS, MERGE ...
- Kromě SQL/XML přibývá Part 11 - SQL/Schemata

SQL 2006

- Kdo nemá XML databázi, není **IN**
- **Jenže** – databáze mají XML typy, umí export/import, ale neumí s XML daty moc pracovat
- Dochází opět poměrně rychle k dohodě a doplnění PART 14 - XML
- Nejde o nový standard, jen o revizi Part 14
- Doplněno hlavně o XQUERY a další funkce pro práci s XML daty

Není nad to se domluvit ...



SQL 2008

- Relativně velká další revize, neobsahuje žádné velké inovace, vzniká spíše na základě nátlaku velkých firem a standardizuje funkčnosti, které již v jsou v databázích implementovány,
 - *např. Oracle má instead of trigger už od 2002*
 - Rozšíření příkazu select limitování výstupu např. `FETCH FIRST 10 ROWS ONLY`
 - Příkazy pro diagnostiku
 - Příkaz `TRUNCATE`
 - `INSTEAD OF` trigger
 - Partitioned `JOIN`
 - Xquery regulární výrazy
 - Definice XML exportu a importu

SQL 2011

- Zatím poslední oficiální vydání
- Existují již corrections 2013 i 2014


- Delete in merge

- Pipellined DML


- Další možnosti
limitace selectů

- Nové tzv. windows

funkce: LAG, LEAD,
Nth_VALUE, First_value,
Last_value



```
select changed_rows.empno  
from new table  
(update emp set salary = 0  
where empno > 100)  
as changed_rows
```



```
SELECT Name, Salary  
FROM Emp  
ORDER BY Salary DESCENDING  
(OFFSET 10 ROWS)  
FETCH NEXT 10 ROWS ONLY  
FETCH NEXT 10 ROWS WITH TIES  
FETCH FIRST 10 PERCENT ROWS  
ONLY
```


Temporal

- Ale hlavně **Temporal features**
- Řeší problém s časovou závislostí dat
- Tyhle problémy jsou tu odjakživa, zabýval se jimi už Ted Codd
- O co jde:
- Problém č. 1:
 - Pro smlouvu s bankou XYZ platí do května všeobecné podmínky verze 1, ale od června už verze 2
 - Šéfem oddělení byl do dubna Petr, ale od května je už šéfem Karel
 - Do března byla ve funkci rektorka od března už rektor
 - Potřebuju, aby se např. při tisku duplikátů dokumentů z dřívější doby automaticky tiskla správná data – v tomto případě jména či znění správných všeobecných podmínek

No jo s bankama jsou pořád nějaký problémy a o rektorkách ani nemluvíme ...



Temporal

- Problém č. 2:
 - Potřebuju dohledat od kdy je šéfem oddělení Karel
 - Potřebuji dohledat, jaké všeobecné podmínky platí pro mou smlouvu s bankou XYZ
 - Chci vědět, od kdy je ve funkci rektor
- Oba problémy se dnes běžně řeší na aplikační úrovni
- SQL 2011 ale definuje tyto Temporal funkčnosti již jako vlastnosti databáze

Takhle jednoduše se to všechno vyřeší ... ?!



Temporal

- Problém č. 1 řeší záznamy s uživatelsky definovanou platností
 - U každého záznamu mám automaticky uloženo od kdy do kdy platí (obě platnosti jsou povinné)
 - Položky mohu vyplnit uživatelsky, default pro *platnost od* je „teď“ pro *platnost do* maximální datum, které daná databáze umí
 - Platnosti umím automaticky zahrnout i do referenční integrity (primární a cizí klíče)

Temporal

- Problém č. 2 řeší záznamy se systémově definovanou platností
 - Při každé změně (INSERT, UPDATE, DELETE) se automaticky v DB uloží i původní verze záznamu
- V obou případech pak mohou být použity pro výběr záznamů některé z operátorů: CONTAINS, OVERLAPS, EQUALS, PRECEDES, SUCCEEDS, IMMEDIATELY PRECEDES, IMMEDIATELY SUCCEEDS.

Temporal

- Oba problémy byly řešeny již v roce 1995 v rámci přípravy SQL 99
- Tehdy byla idea vytvořit nový datový typ „perioda platnosti“ a nadstavby SQL nazvané TSQL2
- Znamenalo to ale modifikaci prakticky všech částí standardu a do toho se nikomu nechtělo, takže tento pokus skončil neúspěchem
- Nyní se řeší **jen** přidáním metadat k tabulce, což ovlivnilo **jen** Part 2: *Foundation*
- A to už prošlo

Temporal

- Example

```
INSERT INTO Emp  
VALUES (22217,  
DATE '2010-01-01',  
DATE '2011-11-12', 3)
```

```
CREATE TABLE Emp(  
ENo INTEGER,  
EStart DATE,  
EEnd DATE,  
EDept INTEGER,  
PERIOD FOR EPeriod (EStart, EEnd))
```

ENo	EStart	EEnd	EDept
22217	1.1.2010	12.11.2011	3

```
UPDATE Emp  
FOR PORTION OF EPeriod  
FROM DATE '2011-02-03'  
TO DATE '2011-09-10'  
SET EDept = 4  
WHERE ENo = 22217
```

ENo	EStart	EEnd	EDept
22217	1.1.2010	3.2.2011	3
22217	3.2.2011	10.9.2011	4
22217	10.9.2011	12.11.2011	3

```
SELECT Edept FROM Emp  
WHERE ENo = 22217  
--AND EStart < DATE '2011-01-01, AND EEnd > DATE '2010-01-01'  
AND EPeriod OVERLAPS PERIOD (DATE '2010-01-01', DATE '2011-01-01')
```

Temporal

- Standard ale dále neřeší, zda příkazy INSERT, UPDATE a DELETE „nepřetržitou kontinualitu“ nebo fungují „natvrdo“

ENo	EStart	EEnd	EDept
22217	1.1.2010	3.2.2011	3
22217	3.2.2011	10.9.2011	4
22217	10.9.2011	12.11.2011	3

Delete from EMP where Edept=4

ENo	EStart	EEnd	EDept
22217	1.1.2010	3.2.2011	3
22217	10.9.2011	12.11.2011	3

ENo	EStart	EEnd	EDept
22217	1.1.2010	12.11.201	3

- Každá implementace to řeší jinak – speciálními klíčovými slovy, které určují, jak se příkaz má chovat

Temporal

- Systémově verzované tabulky – example:

```
CREATE TABLE Emp  
ENo INTEGER,  
Sys_start TIMESTAMP(12) GENERATED ALWAYS AS ROW START,  
Sys_end TIMESTAMP(12) GENERATED ALWAYS AS ROW END  
PERIOD FOR SYSTEM_TIME (Sys_start, Sys_end))  
WITH SYSTEM VERSIONING
```



To né, oni snad o mně budou
vědět úplně všechno?

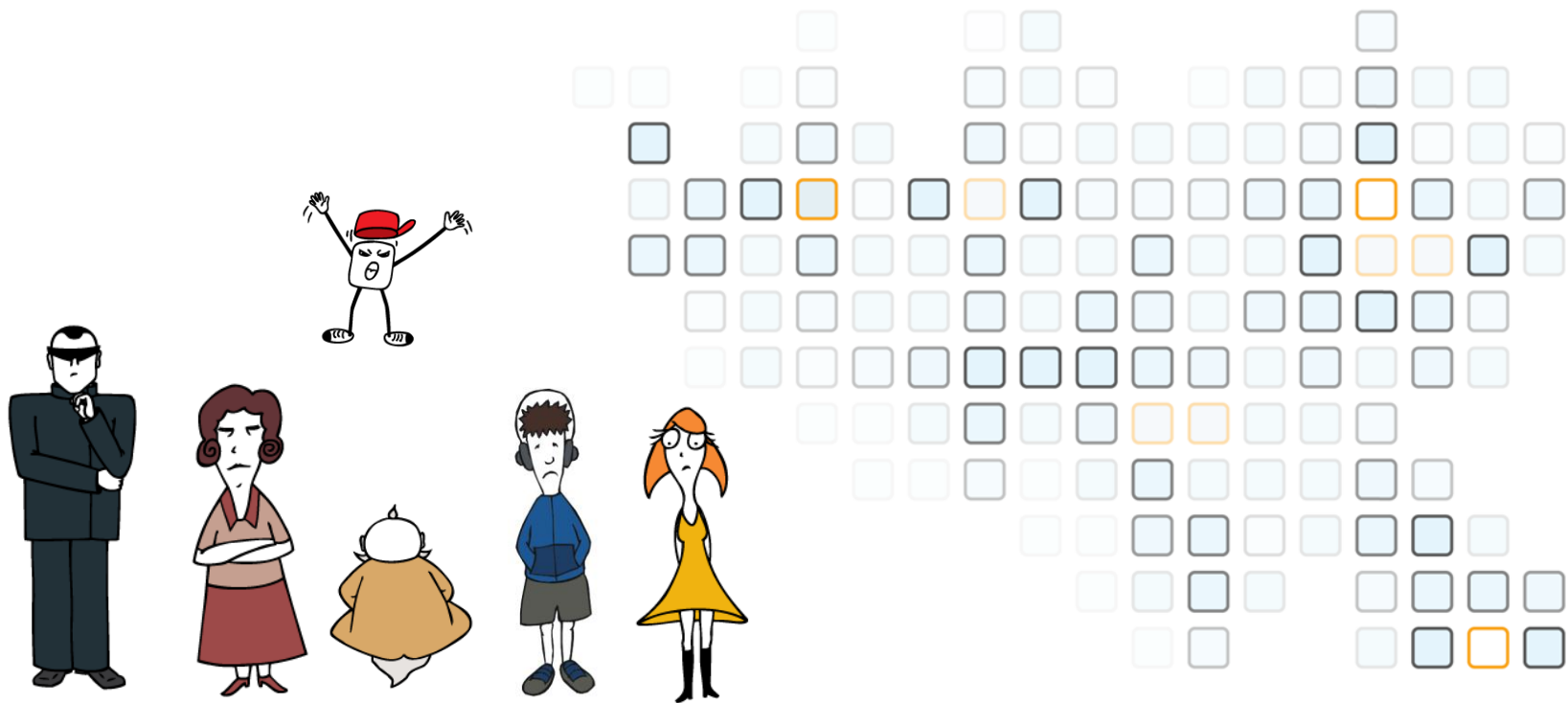
```
SELECT ENo  
FROM Emp FOR SYSTEM_TIME AS OF  
TIMESTAMP '2011-07-01 00:00:00'  
WHERE ENo = 22217
```


Temporal

- Obě funkčnosti lze kombinovat, tj. existují *bitemporal tables*
- Temporal features jsou implementovány v:
 - IBM DB2 10 (jaro 2012)
 - Oracle enterprise i standard 12c (léto 2013)

Jasně, tomu rozumím a co vy?





HURÁ UŽ JE KONEC - DĚKUJI ZA POZORNOST

Dotazy?