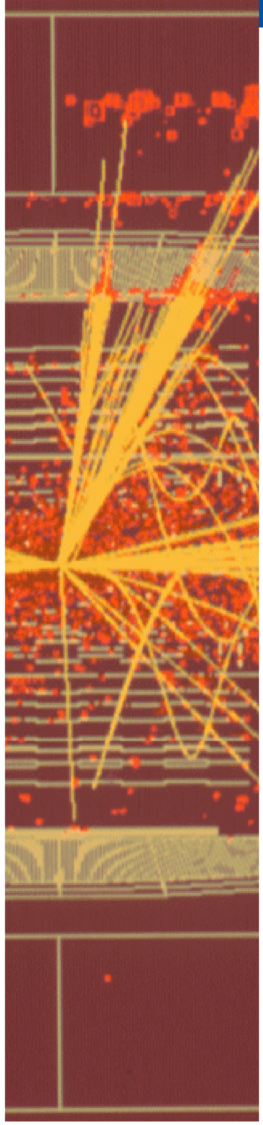
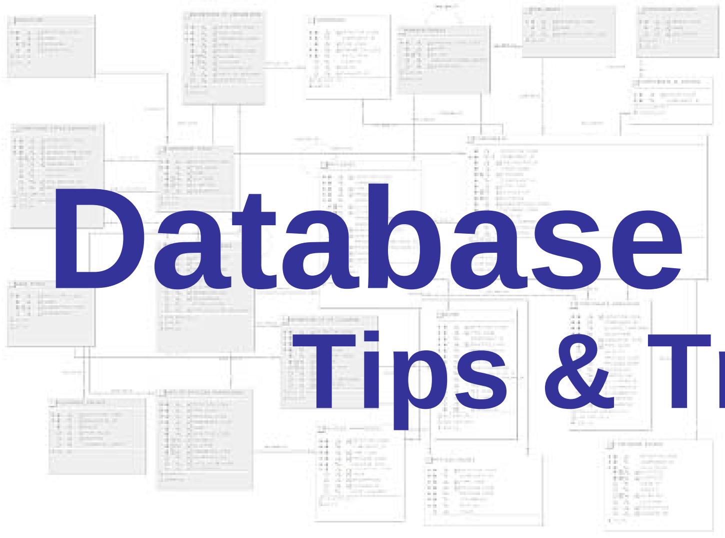
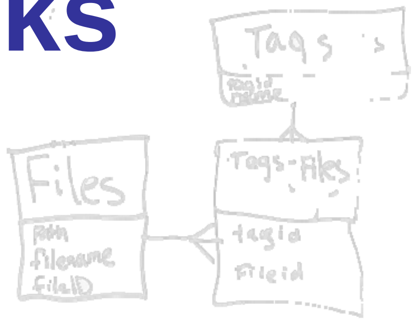
PSS Physics Services Support

CERN IT Department





Database Design
Tips & Tricks



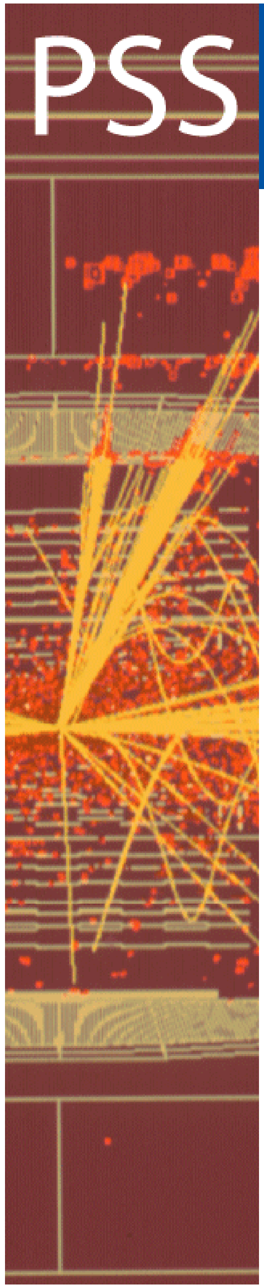
WLCG Service Reliability Workshop

CERN, November 29th, 2007

Dawid Wójcik, CERN IT-PSS

DB Design based on slides by Marta Jakubowska-Sobczak

CERN - IT
Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it



Outline

CERN IT Department

- Database design
- Tips & tricks
  - Indexes
  - Partitioning
  - PL/SQL
- Writing robust applications
- Q&A

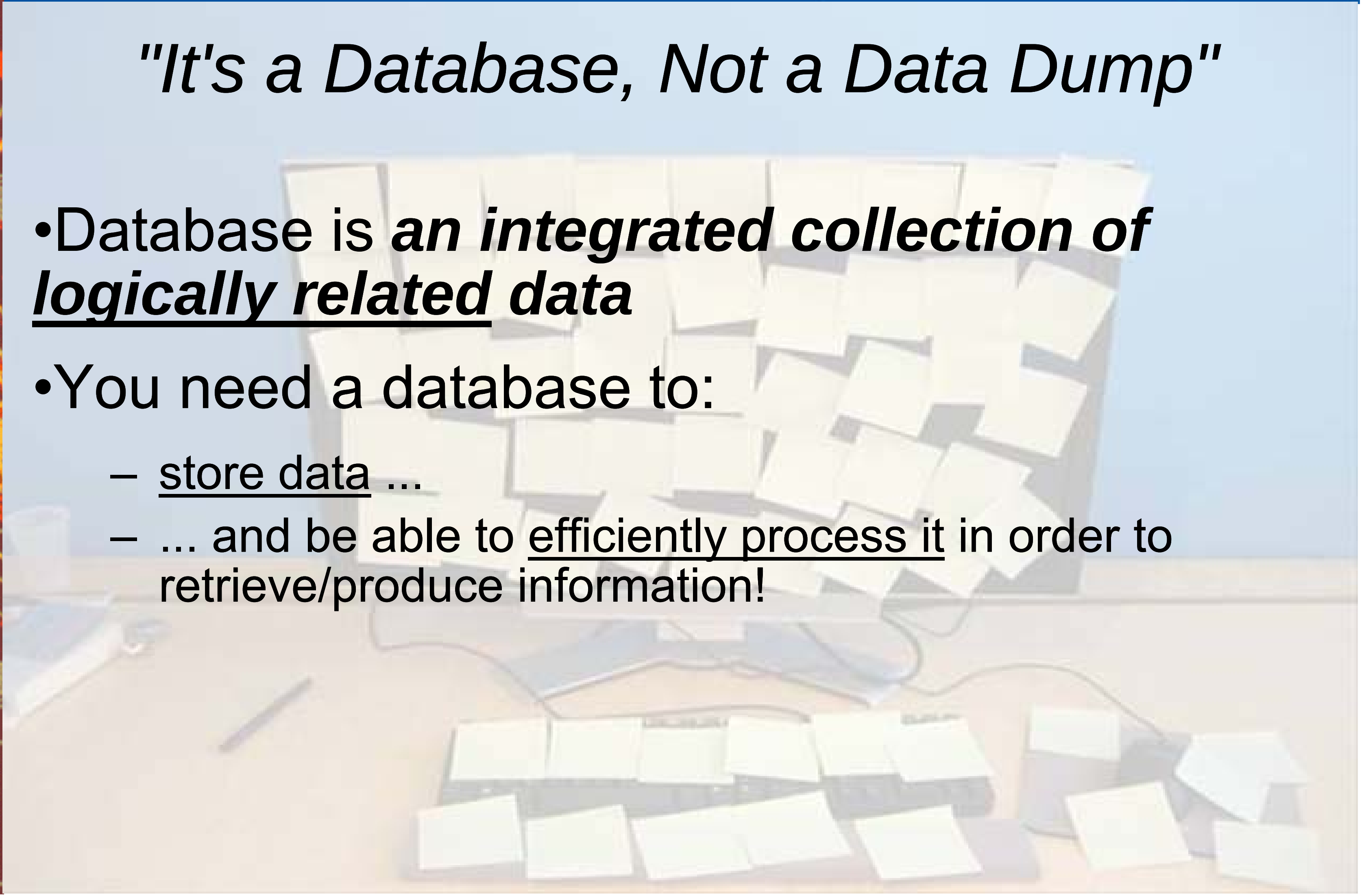
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

PSS

Database design

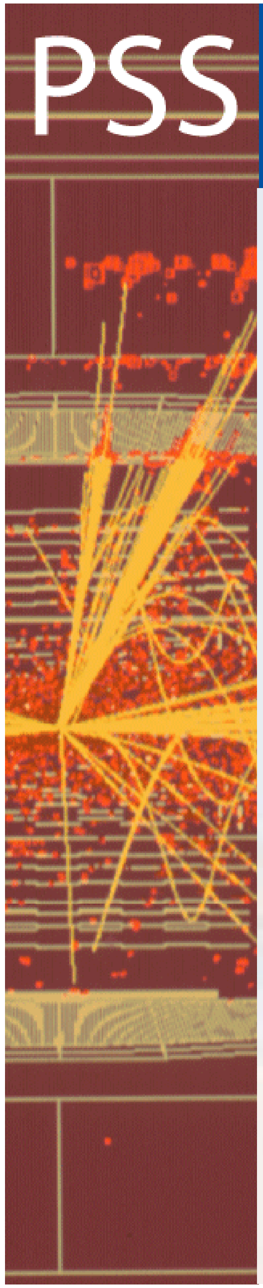
CERN IT Department

"It's a Database, Not a Data Dump"



- Database is ***an integrated collection of logically related data***
- You need a database to:
  - store data ...
  - ... and be able to efficiently process it in order to retrieve/produce information!

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it



Database design – goals

CERN IT Department

"It's a Database, Not a Data Dump"

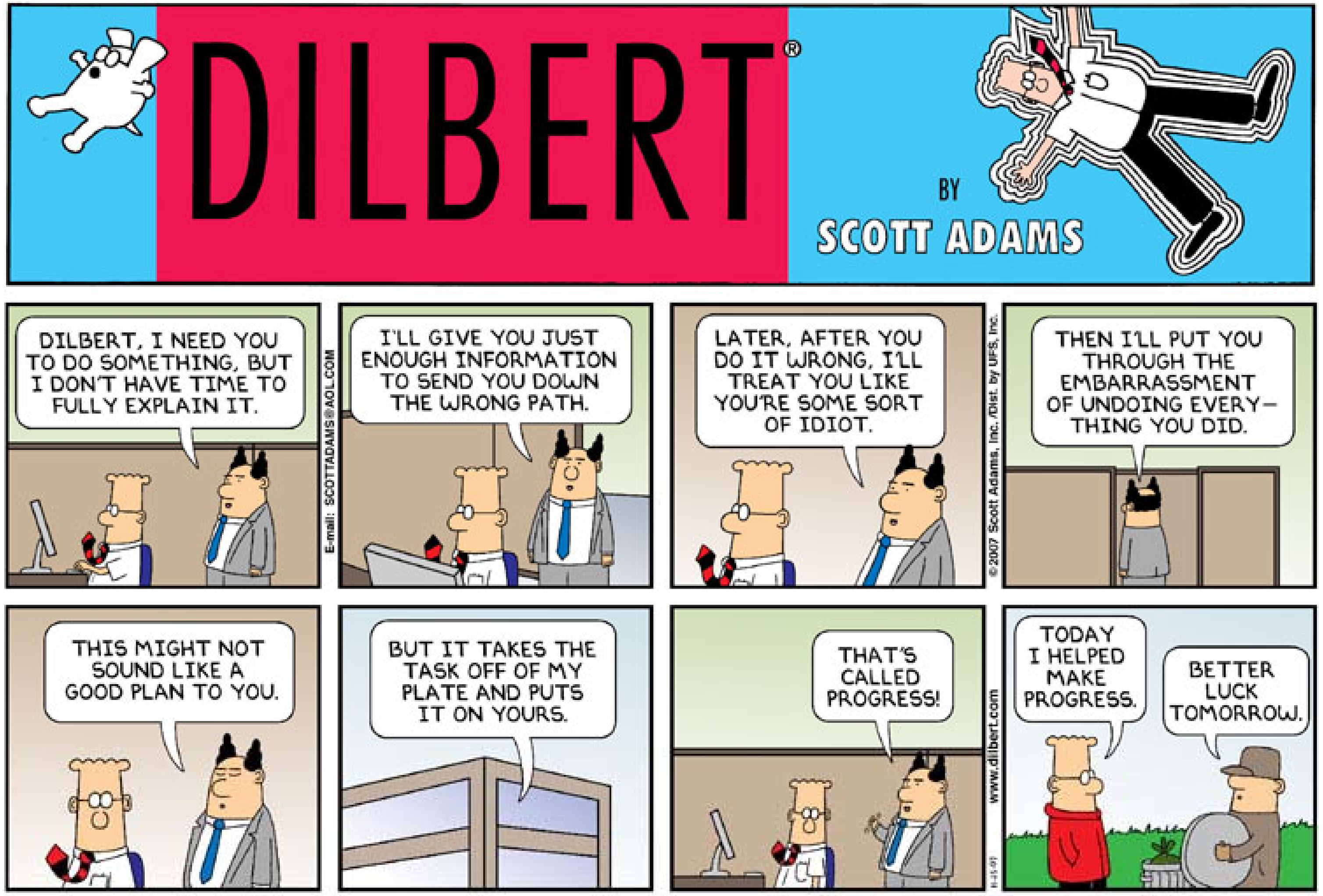
- Database design – define how to store data to:
  - avoid unnecessary redundancy
    - Storage is not unlimited.
    - Redundant data is not logically related
  - retrieve information <u>easily</u> and <u>efficiently</u>
    - Easily – does not necessarily mean with a simple query.
    - Efficiently – using built-in database features.
  - be scalable for data and interfaces
    - **Performance** is in the **design**!
    - Will your design scale to predicted workload (thousands of connections)?

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

PSS

Conceptual design – know your goal

CERN IT Department


DILBERT®
BY SCOTT ADAMS
DILBERT, I NEED YOU TO DO SOMETHING, BUT I DON'T HAVE TIME TO FULLY EXPLAIN IT.
I'LL GIVE YOU JUST ENOUGH INFORMATION TO SEND YOU DOWN THE WRONG PATH.
LATER, AFTER YOU DO IT WRONG, I'LL TREAT YOU LIKE YOU'RE SOME SORT OF IDIOT.
THEN I'LL PUT YOU THROUGH THE EMBARRASSMENT OF UNDOING EVERYTHING YOU DID.
THIS MIGHT NOT SOUND LIKE A GOOD PLAN TO YOU.
BUT IT TAKES THE TASK OFF OF MY PLATE AND PUTS IT ON YOURS.
THAT'S CALLED PROGRESS!
TODAY I HELPED MAKE PROGRESS.
BETTER LUCK TOMORROW.
E-mail: SCOTTADAMS@AOL.COM
© 2007 Scott Adams, Inc./Dist. by UFS, Inc.
www.dilbert.com

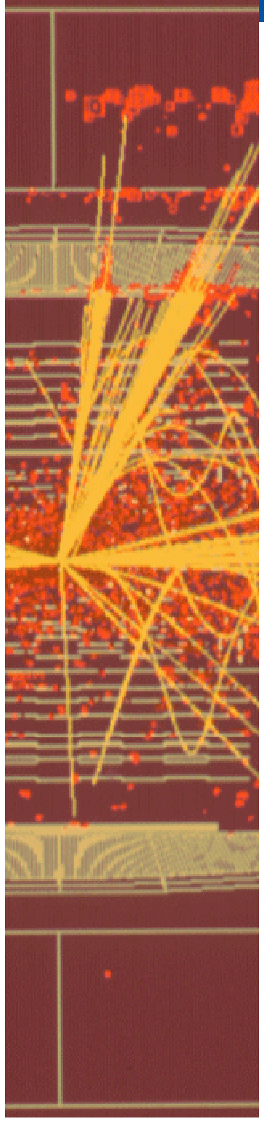
PSS

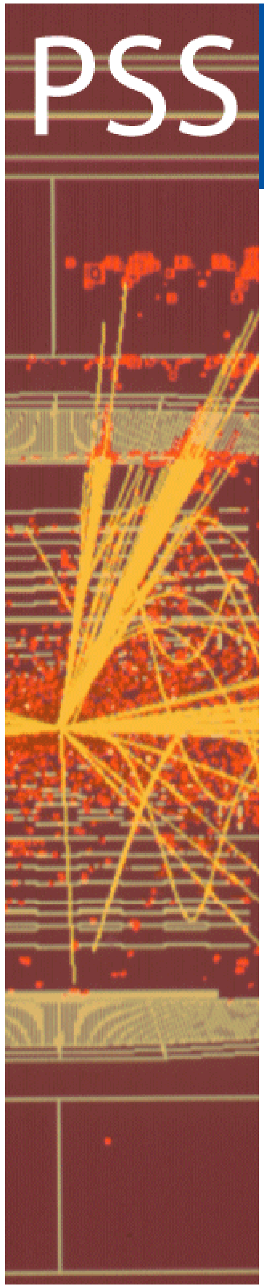
Conceptual design

CERN IT Department

- Conceptual design
  - Process of constructing a model of the information used in an enterprise.
  - Is a conceptual representation of the data structures.
  - Is independent of all physical considerations.

- *Input:* database requirements
- *Output:* conceptual model





Conceptual design – practice

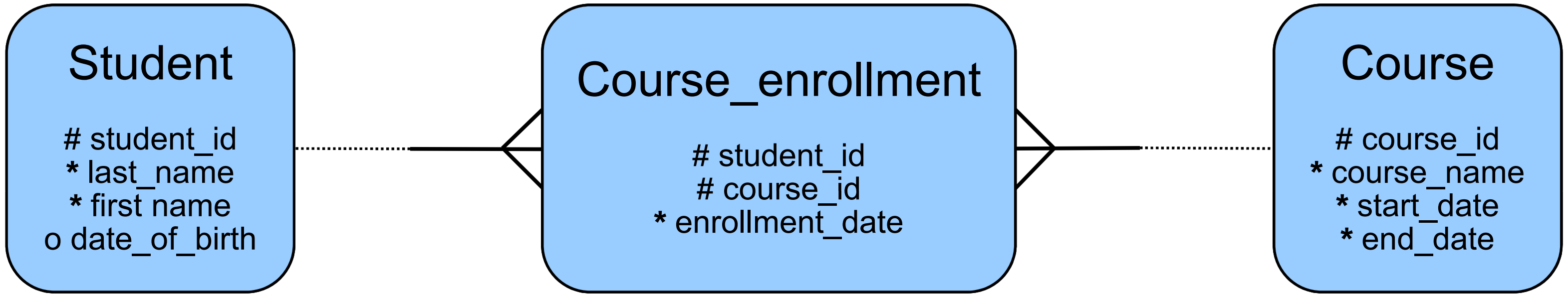
- The Entity-Relationship model (ER) is most common conceptual model for database design:
  - Describes the data in a system and how data is related.
  - Describes data as ***entities***, ***attributes***, and ***relationships***.
  - Can be easily translated into many database implementations.

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Modeling relationships - example

- Many – to – many (M:N)
  - A student can be registered on any number of courses (including zero)
  - A course can be taken by any number of students (including zero)

- Logical model – normalized form:



PSS

Normalization

CERN IT Department

- Objective – validate and improve a logical design, satisfying constraints and avoiding duplication of data.
- Normalization is a process of decomposing relations with anomalies to produce smaller well-structured tables:
  - First Normal Form (1NF)
  - Second Normal Form (2NF)
  - Third Normal Form (3NF)
  - Other: Boyce/Codd Normal Form (BCNF), 4NF ...
- Usually the 3NF is appropriate for real-world applications.

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

First Normal Form (1NF)

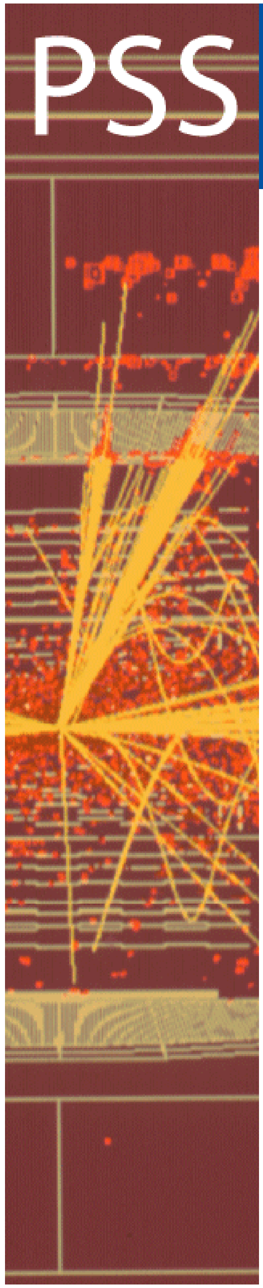
- All table attributes values must be atomic (multi-values not allowed)
  - Eliminate duplicative columns from the same table.
  - Create separate tables for each group of related data and identify each row with a unique column (the primary key).

Manager	Subordinate1	Subordinate2
Helen Smith	John Doe	Marc Brown

Manager	Subordinates
Helen Smith	John Doe, Marc Brown

Manager ID	Subordinate ID
763	6
763	3

Employee ID	Name	Surname
3	Marc	Brown
6	John	Doe
763	Helen	Smith



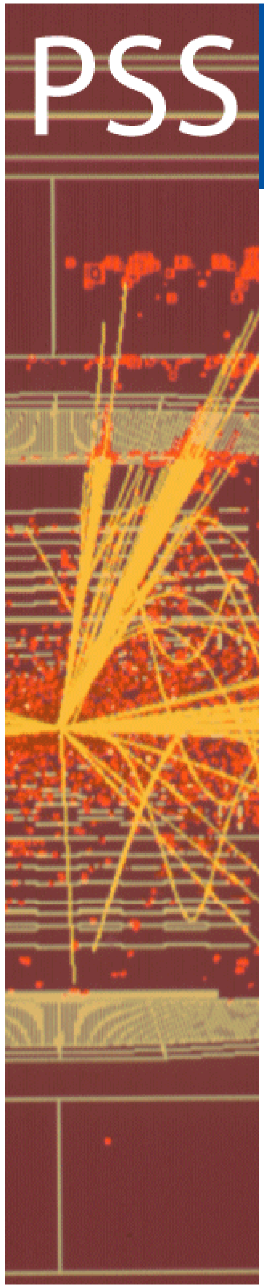
Second Normal Form (2NF)

CERN IT Department

- 1NF
- No attribute is dependent on only part of the primary key, they must be dependent on the entire primary key.

SID	SNAME	CID	CNAME	GRADE
224	Waters	M120	Database Management	A
224	Waters	M122	Software Engineering	B
224	Waters	M126	OO Programming	B
421	Smith	M120	Database Management	B
421	Smith	M122	Software Engineering	A
421	Smith	M125	Distributed Systems	B

Violation of the 2NF!



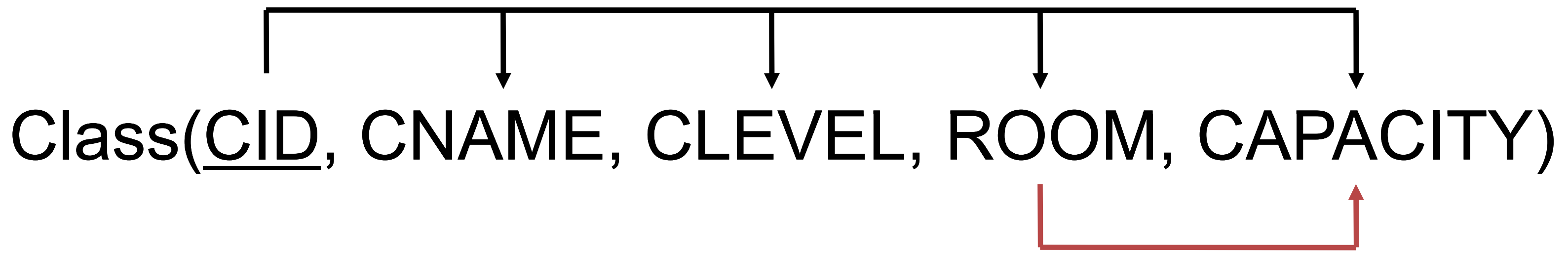
Normalization to 2NF

- For each attribute in the primary key that is involved in partial dependency – create a new table.
- All attributes that are partially dependent on that attribute should be moved to the new table.

$$\begin{tikzcd} & \text{Student}(\underline{\text{SID, CID}}, \cancel{\text{SNAME, CNAME}}, \text{GRADE}) \arrow[dl] \arrow[dr] & \\ \text{Student}(\underline{\text{SID}}, \text{SNAME}) & & \text{Class}(\underline{\text{CID}}, \text{CNAME}) \end{tikzcd}$$

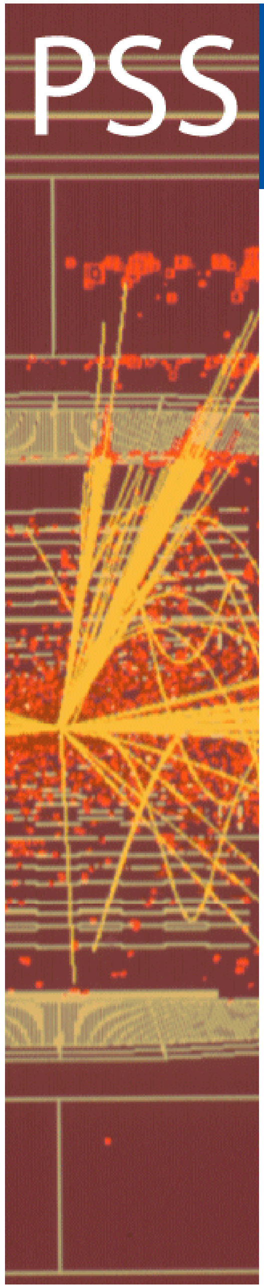
Third Normal Form (3NF)

- 2NF
- No transitive dependency for non-key attributes
  - Any non-key attribute cannot be dependent on another non-key attribute



Class(<u>CID</u>, CNAME, CLEVEL, ROOM, CAPACITY)

Violation of the 3NF!



Normalization to 3NF

- For each non-key attribute that is transitive dependent on a non-key attribute, create a table.

Class(CID, CNAME, CLEVEL, ROOM, CAPACITY)

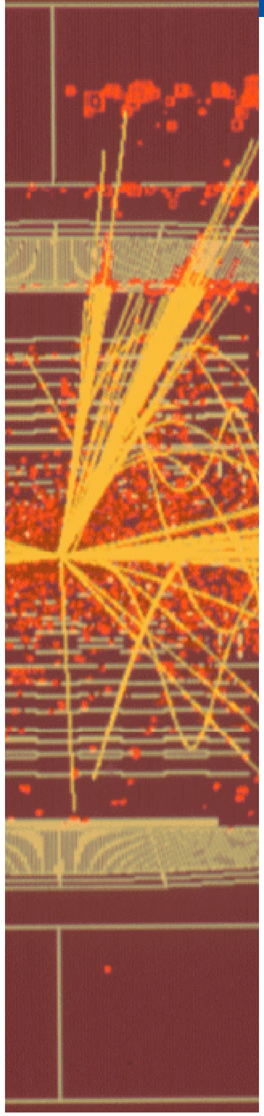
→ Class(CID, CNAME, CLEVEL, ROOMID)

→ Room(ROOMID, CAPACITY)

PSS

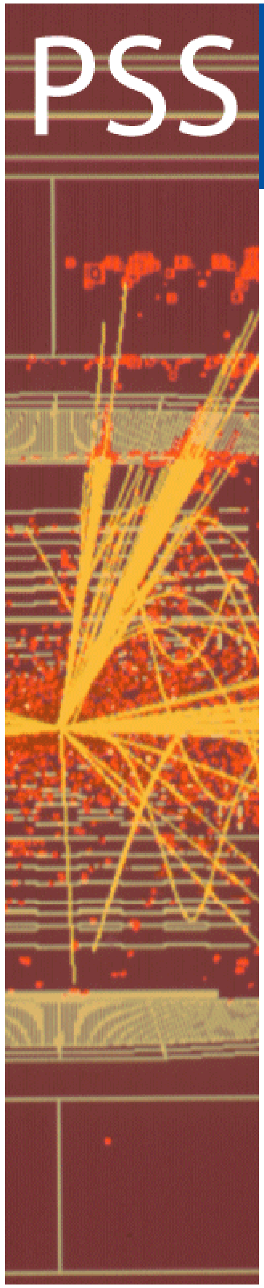
Integrity constraints - PK

CERN IT Department



- Primary keys (PK)
  - Role: Enforce entity integrity.
  - Attribute or set of attributes that uniquely identifies an entity instance.
  - Every entity in the data model must have a primary key that:
    - is a non-null value
    - is unique
    - it does not change or become null during the table life time (time invariant)

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it



Integrity constraints - FK

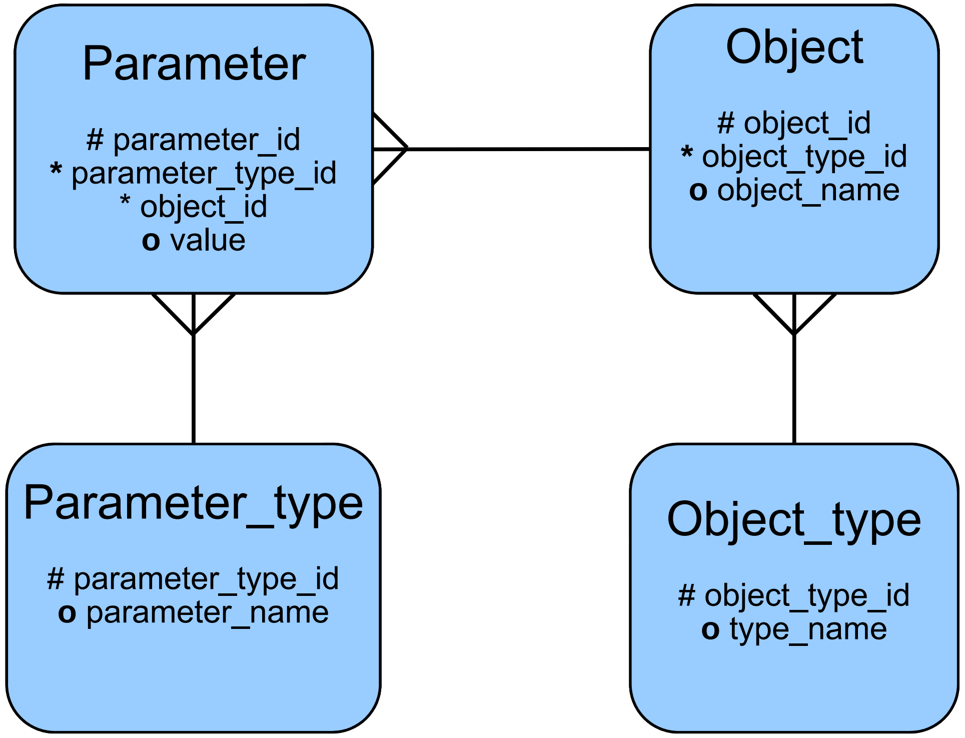
- Foreign keys (FK)
  - Role: maintains consistency between two tables with a relation.
  - The foreign key must have a value that matches a primary key in the other table or be null.
  - An attribute in a table that serves as primary key of another table.
  - Use foreign keys!
    - foreign keys with indexes on them improve performance of selects, but also inserts, updates and deletes.
- Checks
  - NOT NULL

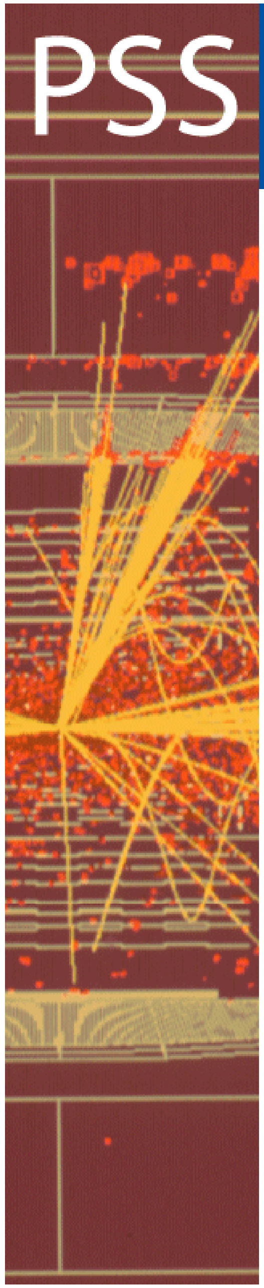
PSS

Schema design – best practices

CERN IT Department

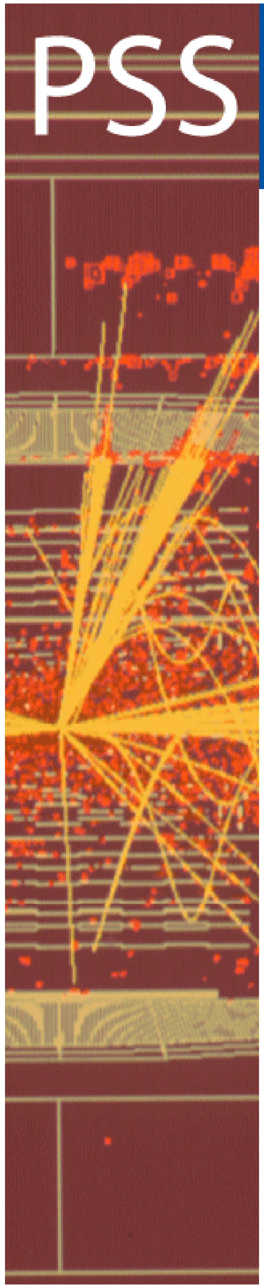
- Use generic structures
  - whenever there's a need of higher flexibility
  - do not use it to model "everything"





Schema design – best practices

- Column types and sizing columns
  - VARCHAR2(4000) is not the universal column type
    - high memory usage on the client
    - it makes data dump, not database
    - you cannot index a key longer than ≈6400 bytes
    - use proper data types
  - Put “nullable” columns at the end of the table



Schema design – best practices

- Estimate future workload
  - read intensive?
  - write intensive?
  - transaction intensive?
  - mixture? – estimate the amount of each type
- Design indexes knowing the workload
  - what will users query for?
    - Minimize number of indexes using proper column order in the indexes.
    - Create views, stored procedures (PL/SQL) to retrieve the data in the most efficient way – easier to tune in a running system.
  - what is the update/insert/delete pattern?
    - Create indexes on foreign keys.

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Tips & tricks

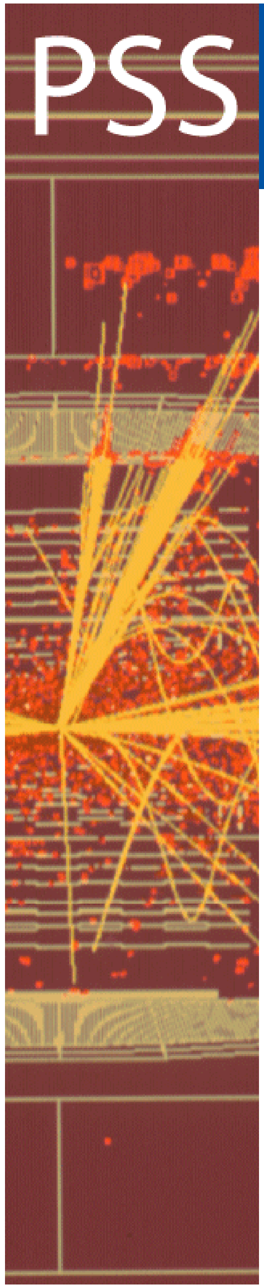
PSS

Indexes – tips & tricks

CERN IT Department

- Selective indexes
  - suppose we have huge table with jobs, most of them already processed (processed_flag = 'Y'), we want only to index non-processed jobs (save index space, improve insert performance):
    - Use bitmap index – very bad idea on frequently updated tables.
    - Create selective view.
    - Create function based index:

```
create index my_indx on tbl_t1
  ( case when processed_flag = 'N' then 'N'
  else NULL
  end);
```



CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

PSS

Indexes – tips & tricks

CERN IT Department

- Reversed indexes
  - suppose we have many concurrent programs that insert into the same table, the table has a primary key column populated by an increasing sequence. There are no range scans on that column. The data is deleted time to time according to some rules which leave some old data undeleted in the table.
    - *Create reversed index:*

```
alter index index_name rebuild reverse;
```

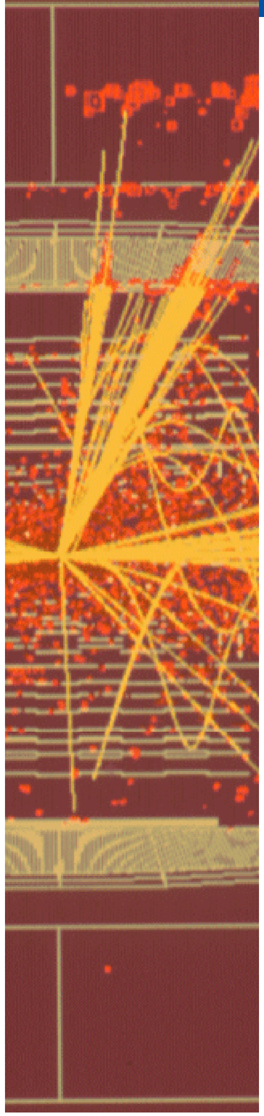
    - Reversed index decreases contention on the index, especially in RAC environment – improves insert/update performance.
    - You can no longer make range scans using reversed index.

PSS

Partitioning – tips & tricks

CERN IT Department

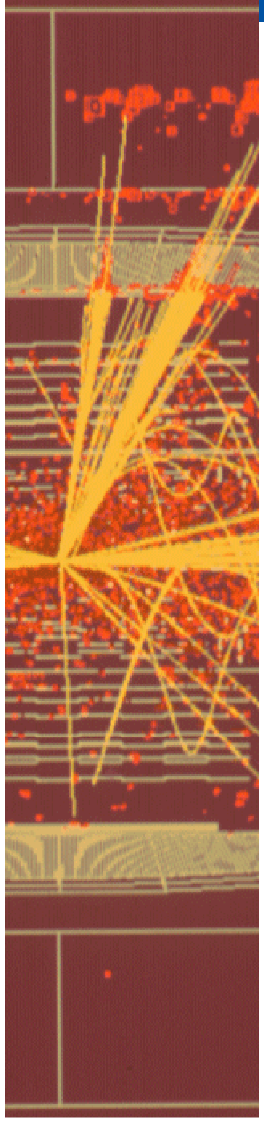
- Investigate partitioning your application
  - You can try partitioning by time, subdetector, subsytem, etc.
  - Benefits:
    - increased availability – in case of loosing one tablespace/partition,
    - easier administration – moving smaller objects if necessary, easier deletion of history, easier online operations on data (ALTER TABLE ... EXCHANGE PARTITION),
    - increased performance – use of local and global indexes, less contention in RAC environment.



CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

PL/SQL – tips & tricks

- Query parse types
  - Hard parse
    - Optimizing execution plan of a query.
    - High CPU consumption.
  - Soft parse
    - Reusing previous execution plan.
    - Low CPU consumption, faster execution.
- Reduce the number of hard parses
  - Put top executed queries in PL/SQL packages/procedures/functions.
  - Put most common queries in views.
  - It also makes easier to tune bad queries in case of problems.



CERN - IT
Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

PSS

PL/SQL – tips & tricks

CERN IT Department

- Reduce the number of hard parses
  - Use bind variables
    - Instead of:

```
select ... from users where user_id=12345
```

    - Use:

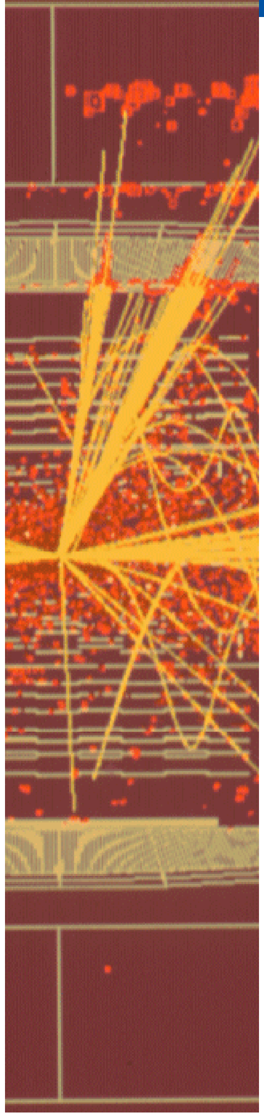
```
select ... from users where user_id=:uid
```

    - Using bind variables protects from sql injection:
      - Before:

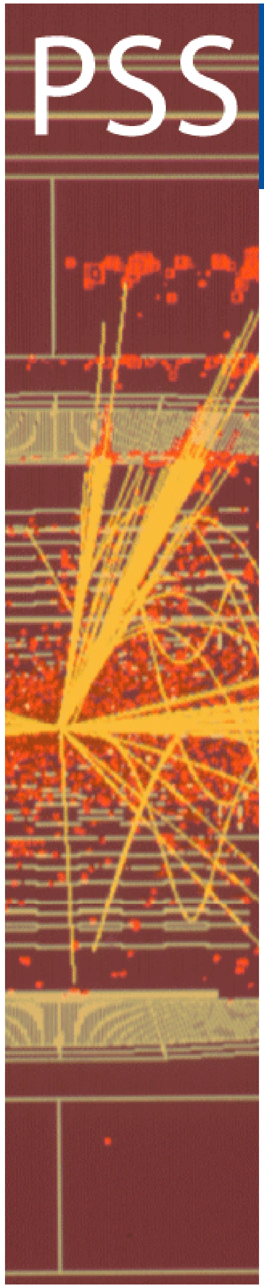
```
sql = "select count(*) from users where
  username='"+user+"' and password='"+pass+'")
 user="hacker"    pass="aaa' or 1=1"
```

      - After:

```
sql = "select count(*) from users where
  username=:user and password=:pass"
```



CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it



PL/SQL – tips & tricks

- Beware of bind variables peeking
  - Optimizer peeks at bind variable values before doing hard parse of a query, but only for the first time.
  - Suppose we have huge table with jobs, most of them already processed (processed_flag = 'Y'):
    - using bind variable on processed_flag **may** change query behavior, depending on which query is processed first after DB startup (with bind variable set to 'Y' or 'N')
  - On a low cardinality column which distribution can significantly vary in time – do not use bind variable only if doing so will result in just a few different queries, otherwise **use bind variables**.

PL/SQL – tips & tricks

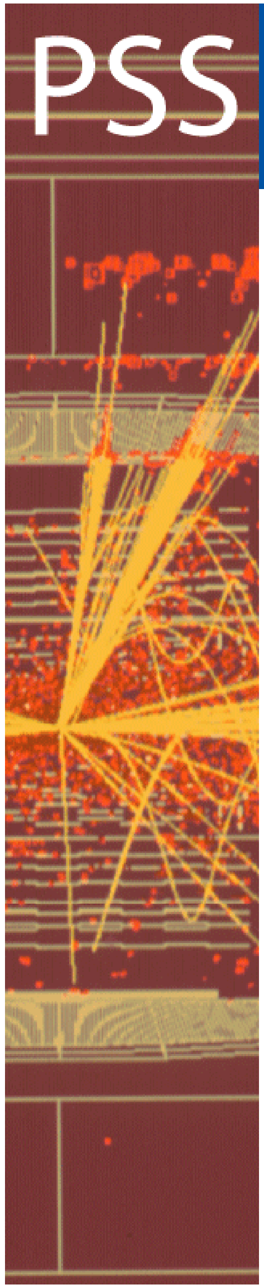
- Reduce the number of hard parses
  - Prepare once, execute many
    - Use prepared statements
    - Dynamic SQL executed thousands of times – consider *dbms_sql* package instead of *execute immediate (?)*
    - Use bulk inserts whenever possible
- Use fully qualified names
    - Instead of:

```
select ... from table1 ...
```

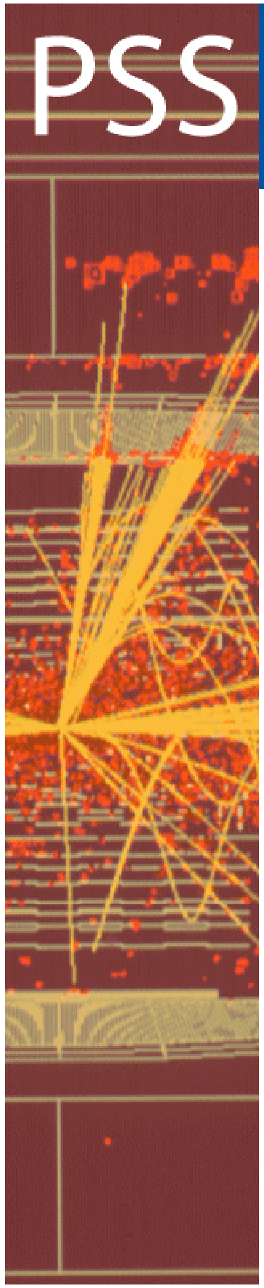
    - Use:

```
select ... from schema_name.table1 ...
```

  - Known bugs – execution in a wrong schema

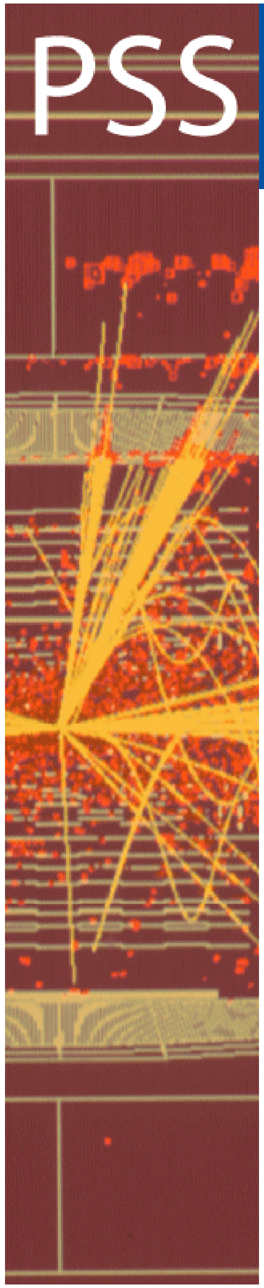


CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it



Writing robust applications

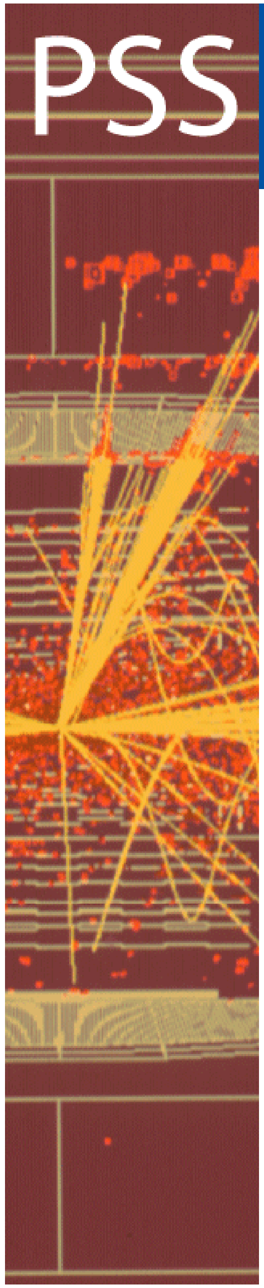
- Use different level of account privileges
  - Application owner (full DDL and DML)
  - Writer account (grant read/write rights to specific objects)
  - Reader account (grant read rights)
  - Directly grant object rights or use roles
    - Caution – roles are switched off in PL/SQL code, one must set them explicitly.
  - Passwords in code get exposed very easily
    - Exposing reader password may result in DoS attacks.
    - Exposing other accounts' passwords may result in data loss.



Writing robust applications

CERN IT Department

- Use connection pooling
  - Connect once and keep a specific number of connections to be used by several client threads (pconnect in OCI)
  - Test if the connection is still open before using it, otherwise try reconnecting
  - Log connection errors, it may help DBAs to resolve any potential connection issues

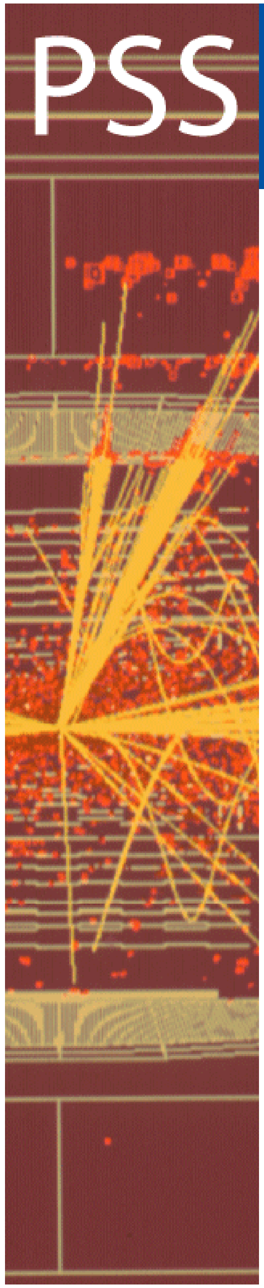


Writing robust applications

CERN IT Department

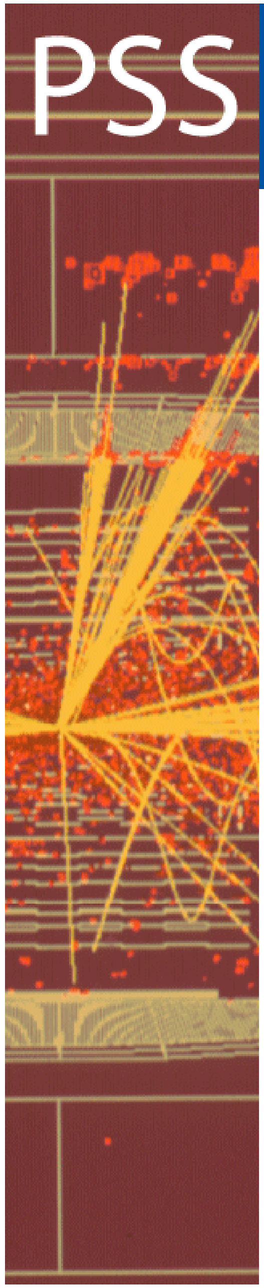
- Error logging and retrying
  - Trap errors
  - Check transactions for errors, try to repeat failed transactions, log any errors (including SQL that failed and application status – it might help to resolve the issue)
  - If you can afford some parts of a transaction (bulk insert) to fail – consider using error logging clause:
    - `CREATE TABLE raises (emp_id NUMBER, sal NUMBER CONSTRAINT check_sal CHECK(sal > 8000));`
    - `EXECUTE DBMS_ERRLOG.CREATE_ERROR_LOG('raises', 'errlog');`
    - `INSERT INTO raises SELECT employee_id, salary*1.1 FROM employees WHERE commission_pct > .2 LOG ERRORS INTO errlog ('my_bad') REJECT LIMIT 10;`
    - `SELECT ORA_ERR_MESG$, ORA_ERR_TAG$, emp_id, sal FROM errlog;`

```
ORA_ERR_MESG$                ORA_ERR_TAG$         EMP_ID SAL
---------------------------- -------------------- ------ -------
ORA-02290: check constraint my_bad                161    7700
 (HR.SYS_C004266) violated
```



Writing robust applications

- Design, test, design, test ...
- Try to prepare a testbed system – workload generators, etc.
- Do not test changes on a live production system.
- IT-PSS provides test and integration system (preproduction) with the same Oracle setup as on production clusters
  - contact PhyDB.Support to obtain accounts and ask for imports/exports.



Physics Services Support

CERN IT Department

Q & A