



**EPAM Systems, RD Dep., RD Dep.**

# **CODING STANDARDS**

---

## 1 NAMING CONVENTIONS

Type	Rules for naming	Examples
Reposi	DA<group abbreviation>_<Name>_<Surname>	DAAU21_Hanna_Petrashka
Pool Request	<module>_<topic>_<task type>	DB_HW SQL_SELECT_HW SQL_SELECT_VerificationTask
Columns (IDENTITY or PK)	<tableName1>ID or <tableName1>_ID	ActorID or Actor_ID
PK	PK_<tableName1>_ <columns separated by _>	PK_actor_actor_ID
FK	FK_<tableName2>_ <columns separated by _>	FK_film_actor_actor_ID

## 2 SELECT STATEMENTS

- Never use SELECT \*. Always specify only the columns needed in a given query.
- Use uppercase letters for all SQL keywords (SELECT, FROM, JOIN, ON, etc.)
- Always specify the explicit schema name for a referenced object (e.g., HR.JOBS) to provide a hint for/clarify the actual location of (database) objects.
- Comment code blocks that are not easily understandable.
  - ✓ Use single-line comment markers (--).
  - ✓ And multi-line comments (/\*.. \*/) for blocking out sections of code
- Verify that all SELECT, INSERT, UPDATE, and DELETE statements have appropriate WHERE clauses (unless it is necessary to operate against an entire table, which should be the exception).
- Always specify ANSI-standard joins (INNER JOIN, LEFT OUTER JOIN).
- Two tables must be joined by FK column(s). Any expressions with '!=', '<>', 'exist/not exist' are not allowed in the JOIN ON clause
- Use meaningful table aliases when joining multiple tables together. Do not use meaningless or arbitrary table aliases, such as a, b, c, d (or some other "incrementing" list of alias names).

*Example: for orders and order\_details, appropriate aliases would be "ord" and "ordet".*
- Do not use single, square brackets or double quotes when using column aliases. Use column aliases when absolutely needed (e.g. to return unambiguous column headers to application code, etc.). When doing so, simply specify the column alias directly using the AS syntax. Example: column\_name1 AS c\_name1. When including aggregates in a result set, include the aggregate keyword as part of the column alias.

*Examples: item\_count, order\_total\_average.*
- Be consistent with indentation - use either tab or space. Preferably starting most SQL keywords at a new line. Use tabs, not a fixed amount of spaces, for indenting all text.
- Based on all these guidelines, a typical query might look like this:

```
SELECT column_name1,
       column_name2
FROM bob.table_name1 tn1
JOIN bob.table_name2 tn2 ON tn1.key_column1 = tn2.key_column1
WHERE tn1.name = 'John'
ORDER BY column_name1;
```

- Comma in SELECT section is always placed after the field name

*Example:*

```
SELECT column_name1  AS c1,
       column_name2  AS c1,
       column_name2  AS c1
```

- All field names in SELECT should be separated by section accordingly to their placement in the report

*Example:*

```
SELECT
    --Report header--
    fc.fiscal_year,
    fc.fiscal_period,
    fc.fiscal_week,

    --Account information--
    a.description + '.' + c.description AS cc_full_account_description,
    st.service_type                     AS service_type_service_type_id,
    cc.concept                          AS cc_concept_id,
    cc.event                            AS cc_event_id
```

- All Aliases, Signs ( =, <, >, !=, etc.), JOIN, ON, etc. must be aligned by columns.

*Example:*

```
SELECT column_name1  AS c1,
       column_name2  AS c2,
       column_name3  AS c3
FROM bob.table_name1 tn1
JOIN bob.table_name2 tn2 ON tn1.key_c1 = tn2.key_c1
LEFT JOIN bob.table_name3 tn3 ON tn2.key_c2 = tn3.key_c2
WHERE tn1.name = 'John' AND
      tn3.department != 'Sale';
```

- If JOIN contains SELECT subquery, this SELECT must comply with the standards.

*Example:*

```
JOIN (
    SELECT msj.meal_sales_header_id,
           SUM(CASE WHEN msj.gl_template_id IN (152) THEN msj.amt ELSE 0 END) AS
           total_sales_amt1,
           SUM(CASE WHEN msj.gl_template_id IN (162,163) THEN msj.amt ELSE 0 END) AS
           total_sales_amt2
    FROM gff.meal_sales_journals msj
    GROUP BY msj.meal_sales_header_id
) AS msj ON msh.meal_sales_header_id = msj.meal_sales_header_id
```

- All filters in WHERE section must be aligned with columns too. Logical operators (AND, OR) should be placed at the end of row.

*Example:*

```
WHERE
    (msh.transaction_date >= @start_date OR @start_date IS NULL) AND
    (msh.transaction_date <= @end_date OR @end_date IS NULL) AND
    (fc.fiscal_year = @fiscal_year OR @fiscal_year IS NULL) AND
    (fc.fiscal_period = @fiscal_period OR @fiscal_period IS NULL) AND
    (fc.fiscal_week = @fiscal_week OR @fiscal_week IS NULL) AND
    (@include_unposted = 1 OR msh.transaction_status not in ('D', 'R'))
```

- Do not use column numbers in the **ORDER BY** clause. Use appropriate column name.

**Incorrect:**

```
SELECT column_name1,
       column_name2
FROM bob.table_name1 tn1
JOIN bob.table_name2 tn2 ON tn1.key_column1 = tn2.key_column1
WHERE tn1.name = 'John'
ORDER BY 2;
```

**Correct:**

```
SELECT column_name1,
       column_name2
FROM bob.table_name1 tn1
JOIN bob.table_name2 tn2 ON tn1.key_column1 = tn2.key_column1
WHERE tn1.name = 'John'
ORDER BY column_name2;
```