	RQ1
DBMS	MySQL is used in 55.9% of projects with DBMS, followed by PostgreSQL (46.0%) and H2 (44.6%).
	Out of the 50 DBMSs analyzed, Interbase, FileMaker, and Virtuoso-SQL were not used in any project.
	160 projects showed no evidence of using any of the 50 DBMS searched in our corpus.
Models	We found evidence of the use of 22 of the 25 relational databases (88.0%) we searched for.
	We found evidence of use of all of the 25 non-relational databases we searched for (100%).
	Relational DBMSs were used in 82.2% of the projects, non-relational DBMSs in 68.3%, and multi-model DBMSs (e.g., Ignite, Virtuoso) in 4.0%.
	Relational DBMSs are more popular than non-relational DBMSs.
	We found that 103 projects (51.0%) adopt both relational and non-relational models.
Domains	The most used DBMSs among the 5 most popular application domains in our corpus are MySQL, PostgreSQL, H2, Oracle, and Redis.
	MySQL is the most used DBMS in the Software Development, Data Management, and Infrastructure Management domains.
	Redis is the leading non-relational DBMS, widely used across key domains.
	MySQL and Redis are frequently employed within the same domains.
	A few DBMSs have high usage rates, while many others are rarely used across the top 5 domains.
Stability	**RQ2**
	We found evidence of 46 DBMSs being adopted at some point in the lifespan of 234 out of the 362 projects of our corpus
	32 projects in our corpus stopped using a DBMS or switched to one not included in our list of 50 DBMSs.
	MySQL is present in 137 projects, followed by H2 in 114 projects and PostgreSQL in 102 projects.
	The top 5 DBMSs are the same as in RQ1, with H2 and PostgreSQL switching positions.
	The analysis shows 22 relational DBMSs, 25 non-relational DBMSs, and one multi-model DBMS (Ignite). This highlights the growing trend of non-relational DBMS adoption in Java projects.
	55.6% of the projects with DBMS usage adopt both data models, while 44.4% adopt only one.
	Interbase and Virtuoso are not very popular DBMSs
	In the Data Management domain, 40 projects used both relational and non-relational DBMSs, 13 used only relational, and 6 used only non-relational DBMSs, showing the highest adoption of both models together.
Migration Patterns	**RQ3**
	Of the projects using MySQL, 41 (29.9%) removed it initially, 20 reintroduced it, and 4 removed it again, leaving MySQL in 112 projects in the final analysis.
	The removals happened for the majority of the DBMSs we surveyed.
	Some DBMSs were never removed once adopted, such as ClickHouse, Microsoft Azure Cosmos DB, and SingleStore.
	We found 296 replacement patterns across 67 projects.
	89 replacement patterns were identified among Relational DBMSs (30.1% of the replacement patterns) across 44 projects (65.7% of the projects with DBMS replacements.
	We found 66 replacement patterns among non-relational DBMSs (22.3%) across 21 projects (31.3%).
	We found 141 replacement patterns involving both models (47.6%) across 43 projects (64.2%).
	In 5.8% of the projects with replacements, the replacements only occur between DBMSs that follow the same data model.
	In 35.8% of projects with DBMS replacements, the change occurs between systems using the same data model, suggesting data integrity as a key factor in replacement decisions. Only 12.5% of replacements involve non-relational DBMSs.
	In 50.0% of the patterns, the replacements occur among distinct data models such as Cassandra replacing PostgreSQL in 3 projects.
	HyperSQL was replaced 23 times in 19 projects. In eight of those times, it was replaced by Redis.
	MySQL was removed 17 times in 13 projects, but it was replaced by a varied set of DBMSs.
	HBase is the second most replaced DBMS (18 times in 5 projects). In four of those times, it was replaced by HyperSQL.
	HyperSQL is the DBMS mostly susceptible to replacements during the projects' history, since it is replaced in 19 different projects.
	Most replaced DBMSs have experienced more than one replacement in certain projects.
	Redis replaced other DBMSs 21 times across 22 projects.

	Oracle replaced other DBMSs 21 times across 19 projects.
	PostgreSQL replaced other DBMSs 20 times across 24 projects.
	RQ4
Initial of the Projects Lifespan	We found MySQL and PostgreSQL in 21 projects, HyperSQL and MySQL in 20, and MySQL and H2 in 17.
	No combinations involving non-relational DBMSs were observed in the initial phase of the projects.
	MS SQL Server → IBM DB2 has the highest lift we found early in the project history (12.88).
	There is a strong association between MS SQL Server and SQLite, with a lift of 10.74. The confidence of this association is 55% in the direction from MS SQL Server to SQLite and 41% in the opposite direction.
	Informix shows strong dependency relationships with PostgreSQL (diff 0.79) and HyperSQL (diff 0.85), indicating a higher likelihood of being adopted alongside these DBMSs.
Middle of the Projects Lifespan	At the start of the projects, we observed co-occurrences involving 10 DBMSs, with the most common combinations being MySQL and PostgreSQL, HyperSQL and MySQL, and MySQL and H2.
	Halfway through the history of the projects, the amount of combined use of DBMSs more than doubled when compared to the beginning,
	MySQL and PostgreSQL, MySQL and H2, and MySQL and Oracle remain among the most used combinations, significantly increasing to 68, 54, and 48 projects, respectively.
	We also highlight the increase in the frequency of combinations involving MS SQL Server halfway through the history of the projects.
	The emergence of combinations involving Cassandra and Redis on 18 projects.
	SapHana → Firebird has a lift of 20.53 and 71% confidence in this direction (Conf(A→B) = 0.71), and 62% confidence in the opposite direction (Conf(B→A) = 0.62).
	Neo4J increases the likelihood of adopting CouchBase by almost 18 times. Notably, there is no difference (Diff = 0.00) in confidence for both directions of this rule (Neo4J → CouchBase).
End of the Projects Lifespan	By the middle of the project history, the number of DBMS co-occurrences increased to 23, with nine involving non-relational DBMSs, including Redis and MySQL, MongoDB and H2, and Cassandra and MySQL.
	Halfway through the history of the projects, there are 9 combinations of non-relational and relational DBMS.
	The synergy between DBMSs increases at the end of the project history.
	MySQL, PostgreSQL, Oracle, and H2 remain among the most frequent combinations throughout the projects' history.
	HyperSQL lost its position to Redis, the combinations involving this DBMS remain popular in various projects.
	Combinations involving Redis became more common, with Redis appearing alongside 22 different DBMSs in various projects.
	Combinations involving non-relational DBMSs increased in frequency from the middle to the end of the analyzed project histories.
	We discovered two rules with 100% confidence: Ingres → Firebird, and InfluxDB → Cassandra.
	The confidences of the rule InfluxBD → Cassandra have a significant difference (Diff = 0.73), indicating a strong dependency of InfluxDB on Cassandra.
	The rule MS SQL Server → SQLite has a lift of 10.74 and confidences of 55% (A → B) and 50% (B → A) at the beginning of the project history. By the middle, the lift drops to 2.80, and A → B confidence decreases to 26%. At the end, the lift slightly decreases to 2.46, with A → B confidence falling to 12%.
	The rule SAP Adaptive Server → Informix, found in the middle of the project history, initially has a lift of 13.80 with 60% confidence in both directions. By the end of the project, the lift increases to 17.32, with 75% confidence for A → B and 60% for B → A.
	RQ5
ORM	We found evidence of ORMs usage in 241 projects.
	MyBatis is the most popular ORM, appearing in 79.6% of projects. Hibernate ranks second, present in 50.6% of projects, followed by Spring, found in 26.5% of the projects.
	The median of the percentage of files related to the ORMs EclipseLink, Hibernate, MyBatis, Spring, JOOQ, and JPA are close and vary from 0.10% to 0.29%.
	JPA and JOOQ are more verbose than the others.
Database-related Files	For the DB-Code files that are of the ".xml" type, the median is 0.17%.
	For the ".java" type, the median is 0.21%, indicating that only a small portion of projects are related to database access functions.
	For files that depend on or use Java DB-Code files, the median is 0.81%, four times higher than the median for the DB-Code files themselves.
	The median in the Total-DB files is 0.80%.
Queries	We found 256 projects that use only SQL, while Builders are used in 105 projects.
	Almost all projects that use Builder also use SQL.
	Fifty-four projects use relational DBMSs that are not among the 25 relational DBMSs we investigated.

	We analyzed the number of files containing SQL queries per project, finding values ranging from 0 to 8,269, with an average of 123.2 files and a standard deviation of 653.9.
	Projects have between 0 and 389 files related to Builders, with an average of 51.5 files.