| **Компетентності рівня junior для Інженерії Даних** | **Загальна оцінка компетентності (max. 12)** | **Компанії** | | | |
|---|---|---|---|---|---|
| | | **Avenga** | **GlobalLogic** | **Quantum** | **DataArt** |
| **1. Обробка Big Data (batch processing) за допомогою Apache Spark** | | | | | |
| **1.1. Apache Spark як фреймворк для обробки даних.**<br>Знає основні поняття та загальне призначення Apache Spark. Розуміє архітектуру Apache Spark, включаючи ролі драйвер-вузла, вузлів обчислення та менеджера кластеру/ресурсів. Усвідомлює як працює модель масивних паралельних обчислень у Spark та концепцію "лінивих" розрахунків у Spark (Lazy Evaluation) | **9** | 3 | 1 | 2 | 3 |
| **1.2. Конфігурація та розширення робочого середовища в Apache Spark**<br>Знає основні принципи конфігурації завдань у Apache Spark та механізми розширення робочого середовища Spark додатковими модулями та пакетами. Розуміє важливість сумісності версій пакетів з версіями Spark, Scala та Hadoop. Усвідомлює принцип налаштування робочого середовища на рівні всього кластера, при якому стандартні методи встановлення пакетів (наприклад, через pip або Maven) не підходять для розподіленого середовища Spark, оскільки пакети мають бути доступні на всіх вузлах кластера. Вміє підбирати правильні версії пакетів, щоб вони були сумісні з версіями Spark, Scala і Hadoop, які використовуються в кластері. | **3** | 0 | 1 | 1 | 1 |
| **1.3 Дії та трансформації в Apache Spark**<br>Знає основи lazy evaluation у Spark. Ідентифікує, які операції є діями, а які трансформаціями. Розуміє основні концепції життєвого циклу програми обробки даних у Spark: побудова DAG (Directed Acyclic Graph), створення логічного плану, створення фізичного плану, динамічна компіляція байт-коду, виконання. Здатний писати код, що мінімізує пересилання великих обсягів даних на драйвер-вузол, щоб уникнути помилок Out Of Memory (OOM) та усвідомлює ризики виокнання дій, пов'язаних з цим пересиланням (операція collect). | **6** | 1 | 1 | 2 | 2 |
| **1.4 Операція Join та її стратегії у розподілених обчисленнях**<br>Знає основні типи джойнів: лівий (left join), правий (right join), внутрішній (inner join), зовнішній (outer join), картезіанський добуток (cross join). Знає особливості та призначення різних стратегій джойнів: Broadcast Join, Hash Join, Sort-Merge Join. Усвідомлює наслідки у вигляді навантаження системи та додаткового шафлінгу (shuffling) даних у кластері внаслідок використаннія Join. Вміє використовувати різні стратегії Join у Spark в залежності від розміру та характеру даних. | **8** | 2 | 2 | 2 | 3/2 |

| | | | | | |
|---|---|---|---|---|---|
| **1.5. Partitions в розподілених обчисленнях**<br>Розуміє, що таке partitions і яку роль вони відіграють у розподілених обчисленнях, основні причини їх використання (паралельна обробка даних і покращення продуктивності). Здатний виконувати перерозподіл даних (repartitioning) та коалесценцію (coalesce) для оптимізації використання ресурсів. Розуміє, що перерозподіл партішенів (repartitioning) може призводити до перемішування (shuffling) даних, що є відносно дорогою операцією. Усвідомлює важливість правильного налаштування кількості партішенів для балансування навантаження та уникнення вузьких місць (bottlenecks) у процесі обробки даних. | **7** | 1 | 1 | 3 | 2 |
| **1.6 Кешування та збереження проміжних результатів у Apache Spark**<br>Знає основні цілі використання кешування та збереження проміжних результатів у Spark, зокрема у випадках повторного використання DataFrame у різних гілках обробки даних. Розуміє механізм чекпоінтів (checkpoints) та його основне призначення. Ідентифікує різні рівні зберігання проміжних даних: в оперативній пам'яті, на диску та змішаних. Здатний використовувати кешування (cache) та персистенцію (persist) для зберігання проміжних результатів у Spark. | **5** | 1 | 0 | 2 | 2 |
| **1.7 Принципи превентивного покращення швидкодії в Apache Spark**<br>Знає базові принципи оптимізації продуктивності обробки даних у Spark.<br>Вміє зменшувати обсяг даних перед виконанням джойну для зменшення навантаження на систему.<br>Усвідомлює необхідність уникати використання UDF (User Defined Functions).<br>Здатний виконувати репартішен по колонці, яка часто використовується в Join, для оптимізації процесу.<br>Застосовує broadcast join, коли маленький набір даних з'єднується з великим, для зменшення обсягу шафлінгу.<br>Замінює патерн "group by + join" на "window functions", коли це можливо, для підвищення продуктивності. | **8** | 2 | 0 | 3 | 3 |
| **1.8 Основні операції над даними у Apache Spark**<br>Знає основні операції над даними у Spark (селект, перейменування колонок, додавання колонок, фільтрування).<br>Використовує стандартні функції для трансформації даних (абсолютне значення, максимальне значення тощо)<br>Здатний виконувати операції з'єднання даних (джойн), групування (group by), віконні функції (window functions), юніон, півот і виконання "сирого" SQL запиту.<br>Вміє трансформувати текстовий опис задачі в скрипт, що використовує наведені вище операції. | **10** | 3 | 1 | 3 | 3 |
| **1.9 Дихотомія операційної та датаінженерної систем у контексті Data Engineering**<br>Знає основні поняття операційної системи у контексті Data Engineering, яка виконує операційні задачі бізнесу (наприклад, реєстрація користувачів, створення ордерів тощо). Усвідомлює поняття датаінженерної системи, де відбувається обробка даних без прямої взаємодії з кінцевим користувачем продукту. | **9** | 2 | 1 | 3 | 3 |
| **2. Інструменти та методи розробки й відлагодження для Data Engineering** | | | | | |
| **2.1 Ноутбуки Jupyter або Zeppelin як інструментів для досліджень**<br>Має практичні навички використання ноутбуків Jupyter або Zeppelin для досліджень, створення презентацій та надання прикладів роботи системи, розуміє різницю в структуруванні коду в ноутбуці та в продакшн-коді. Здатний трансформувати дослідження, проведені в ноутбуці, в робочий код та навпаки. | **9** | 3 | 1 | 3 | 2 |
| **2.2 Доступ до метаінформації про дані**<br>Здатний отримати базову інформацію про дані, таку як розмір, схема та базові статистики. Вміє використовувата відповідні плагіни для інтегрованих середовищ розробки (IDE) або застосовувати команди в Jupyter ноутбуці для отримання метаінформації про дані. | **9** | 3 | 2 | 2 | 2 |
| **2.3 Відлагодження за допомогою ноутбуків**<br>Розуміє основи перетворення продакшн-коду в послідовність комірок у ноутбуці для отримання покрокових результатів роботи пайплайну. Здатний формувати релевантну підвибірку даних для збільшення швидкодії пайплайну та локалізації проблем. | **8** | 2 | 2 | 2 | 2 |

| | | | | | |
|---|---|---|---|---|---|
| **2.4 Інструменти віддаленого відлагодження**<br>Знає основні інструменти для віддаленого відлагодження коду та використовує їх для дебагу коду, запущеного на сторонньому сервері або в Docker-контейнері. | **5** | 2 | 0 | 2 | 1 |
| **2.5 Виявлення та усунення помилок і аномалій даних**<br>Знає типові проблеми з даними, такі як дублювання, пропущені значення, та аномалії. Вміє прокомунікувати проблему, пов'язану з даними, відрізняючи її від локальних фіксів. Здатний частково локалізувати проблему, пов'язану з даними, та приймати рішення щодо її походження: чи була вона створена новим кодом, знаходиться у вхідних даних, або виникла під час реалізації поточного завдання. | **8** | 2 | 1 | 2 | 3 |
| **2.6 Використання інструментів візуалізації пайплайнів**<br>Знає основні можливості та функції SparkUI і Spark History Server та використовує їх для отримання загальних уявлень стосовно перебігу пайплайну. Здатний переглядати інформацію про задачі, включаючи згенеровані під час роботи датасети, DAG представлення задачі, налаштування та конфігурації кластеру на якому була запущена задача. | **6** | 2 | 1 | 1 | 2 |
| **2.7 Отримання інформації про трансформації в пайплайні**<br>Знає і використовує основні інструменти для отримання загальної інформації про датафрейми та їх характеристики (explain, printSchema, describe, show) | **9** | 3 | 2 | 2 | 2 |
| **3. Тестування та якість ПЗ** | | | | | |
| **3.1 Піраміда тестування**<br>Розуміє концепції піраміди тестування. Усвідомлює різницю між модульним та інтеграційним/функціональним тестуванням. | **5** | 1 | 1 | 2 | 1 |
| **3.2 Модульне тестування**<br>Знання базових фреймворки для написання модульних тестів. Здатний ідентифікувати основні потоки обробки даних та створювати відповідні тести для них. Розуміє концепції метрики покриття коду тестами та здатний ідентифікувати непокриті тестами частини коду. Здатний створювати мінімально-достатні "штучні" дані для перевірки певних частин пайплайну. | **8** | 1 | 2 | 3 | 2 |
| **3.3 Інтеграційне тестування**<br>Здатний генерувати мінімально необхідні вибірки даних для інтеграційного тестування на основі описаних властивостей та особливостей їх обробки. Вміє створювати тестові вибірки на основі реальних даних з використанням технік анонімізації (когерентне замінення імейлів, юзернеймів, тощо) | **6** | 1 | 1 | 2 | 2 |
| **3.4 Test-Driven Development (TDD)**<br>Розуміє концепцію розробки через тестування та здатний використовувати її на практиці. Здатний ідентифікувати мінімально необхідний набір даних для роботи розроблюваного модулю та написання відповідних тестів для перевірки модулю. | **6** | 2 | 0 | 2 | 2 |
| **4. Сховища даних** | | | | | |
| **4.1 OLAP vs OLTP**<br>Розуміє відмінності між OLAP та OLTP сховищами даних та здатний Аналізувати вимоги бізнесу та обирати відповідну архітектуру сховища даних. Використовує оптимізацію OLAP сховищ для аналітичних запитів та оптимізацію OLTP сховищ для операційних транзакцій та роботи з великим обсягом невеликих даних. | **10** | 3 | 1 | 3 | 3 |
| **4.2 Типи баз даних**<br>Знає основні типи баз даних (реляційні, ключ-значення, колоночні, документоорієнтовані) та розуміє загальні сфери застосування кожного типу баз даних. Вміє підключатися до різних типів баз даних у середині пайплайну та усвідомлює ризики використання баз даних, що належать операційній системі. Ідентифікує різницю між базами даних, що працюють з рядками та колонками. та розуміє, як кожен тип бази даних може вплинути на роботу пайплайну обробки даних. | **8** | 3 | 1 | 2 | 3/2 |

| | | | | | |
|---|---|---|---|---|---|
| **4.3 Типи файлів та еволюція схеми даних**<br>Знає різні типи файлів (текстові, бінарні, файлові формати з колонками та рядками), розуміє переваги та недоліки кожного типу файлів. Усвідомлює проблему еволюції схеми даних у файлових форматах. Має практичні навички роботи з файлами, що зберігаються в об'єктних сховищах, таких як AWS S3. | **10** | 3 | 1 | 3 | 3 |
| **4.4 Оптимізація сховища даних за допомогою партішенінгу (Partitioning)**<br>Розуміє різницю між партішенінгом в контексті трансформацій у пайплайні обробки даних та в контексті сховища даних. Ідентифікує компроміс між кількістю файлів та їх розміром і знає способи його вирішення. Здатний використовувати партишенінг як предикат попереднього фільтру та усвідомлює його вплив на швидкодію обробки даних. Використовує бакетинг як альтернативний метод реалізації партішенінгу. | **7** | 2 | 1 | 2 | 2 |
| **4.5 Абстракції побудови систем зберігання даних**<br>Ідентифікує основні абстракції систем зберігання даних (Data Warehouse, Data Lake, Lakehouse). Розуміє базові переваги і недоліки кожного типу та сценарії, у яких краще використовувати кожен тип абстракції. Здатний частково нівелювати основні недоліки певного типу організації сховища даних. Розуміє можливості оптимізації роботи з даними через використання відповідних абстракцій зберігання. | **8** | 3 | 1 | 1 | 3 |
| **4.6 Моделі даних**<br>Знає основні моделі даних( Inmon, Star schema, Snowflake model, Wide table), переваги та недоліки кожної моделі. Здатний пояснити принципи організації даних для кожної з моделей. Розуміє, як кожна модель може бути використана для вирішення різних завдань. | **8** | 3 | 1 | 1 | 3 |
| **4.7 Концепція дата-продукту**<br>Знає основні аспекти концепції дата-продукту. Ідентифікує основні характеристики, якими наділяють данні, котрі сприймаються як продукт (метрики якості, наявність метаданих, чітка схема доступу до даних, поліглотність даних). Розуміє необхідність чіткого визначення власника дата-продукту та його орієнтацію на потреби користувача. Усвідомлює, що користувачі можуть бути як внутрішніми, так і зовнішніми, і вміє враховувати їх потреби під час розробки дата-продукту. | **5** | 1 | 1 | 1 | 2 |
| **5. Компліментарні інструменти для обробки та управління даними** | | | | | |
| **5.1 Основи оркестрації даних**<br>Знає основні концепції оркестрації даних та популярні інструменти оркестрації, такі як Prefect і Apache Airflow. Має практичні навички налаштування та використання базових функцій одного з оркестраторів для автоматизації та управління пайплайнами даних. | **8** | 3 | 1 | 1 | 3 |
| **5.2 Використання метаданих платформ**<br>Знає основні функції платформ для перегляду та редагування метаданих, таких як DataHub та подібні інструменти. Розуміє, як метадані стосуються наборів даних та пайплайнів, що їх генерують. Здатний виконувати пошук необхідних наборів даних за допомогою базових фільтрів, а також доповнювати інформацію за допомогою тегів та розширеного опису. Використовує метадані для покращення керованості та організації даних. | **7** | 2 | 1 | 2 | 2 |
| **5.3 Інтеграція та використання датакаталогів**<br>Розуміє основні функції та переваги датакаталогів, принципи організації та управління метаданими у датакаталогах, таких як Hive Metastore та подібні інструменти. Здатний інтегрувати датакаталоги в пайплайни обробки даних. Вміє створювати таблиці в датакаталогах та використовувати створені таблиці в інших пайплайнах обробки даних. | **6** | 2 | 1 | 1 | 2 |

| | | | | | |
|---|---|---|---|---|---|
| **5.4 Знання SQL для маніпуляції даних**<br>Основи SQL: команди SELECT, INSERT, UPDATE, DELETE.<br>Вибірка даних: уміння писати SELECT-запити для отримання даних.<br>З'єднання таблиць: робота з INNER JOIN, LEFT JOIN, RIGHT JOIN.<br>Функції та агрегація: використання SUM, AVG, COUNT тощо.<br>Підзапити та віконні функції: уміння писати складні запити.<br>Керування даними: робота з транзакціями, індексами та створення баз даних, таблиць. | **12** | 3 | 3 | 3 | 3 |
| **6. Потоки даних** | | | | | |
| **6.1 Різниця між потоками даних та чергами**<br>Ідентифікує основні відмінності у підходах до роботи з даними через потоки та черги. Розуміє концепції систем, що пропонують доступ до даних у вигляді потоку подій та черги. Знає інструменти, що забезпечують підтримку обох режимів, та вміє визначати найбільш підходящий режим для конкретного завдання. Здатний працювати зі сховищем даних реального часу та усвідомлює його роль в контексті обробки потокових даних. Знає теоретичні аспекти роботи з такими сховищами та їх вплив на потокову обробку даних. | **8** | 2 | 1 | 2 | 3 |
| **6.2 Обробка потоків даних мікробатчами на нативному рівні.**<br>Розуміє сутність та особливості концепції обробки даних мікробатчами на нативному рівні. Ідентифікує функціональні можливості інструментів, які це підтримують та їх роль в обробці потокових даних. Розуміє основні переваги і недоліки, здатний визначати, коли і в яких випадках цей підхід може бути ефективним та коли краще використовувати інші методи обробки потоків даних. | **6** | 2 | 1 | 1 | 2 |
| **6.3. Використання вікон в потоковій обробці даних**<br>Знає концепцію використання вікон у контексті потокової обробки. Знає різновиди вікон та розуміє відмінності між ними і їх застосування у конкретних випадках. Застосовує вікна для виконання різних операцій з потоковими даними, таких як агрегація, об'єднання потоків, доповнення даних та джойн потоків. | **6** | 2 | 1 | 1 | 2 |
| **6.4 Основи подійно-орієнтованої(event-driven) архітектури**<br>Знає основні ідеї та концепції подійно-орієнтованої (event-driven) архітектури та її застосування як джерела даних для датаінженерної підсистеми. Розуміє, як події генеруються, обробляються та передаються в рамках такої архітектури. | **5** | 1 | 0 | 1 | 3 |
| **7. Розмовна англійська** | | | | | |
| **7.1. Правильна побудова фрази з узгодженням часу**<br>Вміє використовувати відповідні часові форми дієслів і структури речень для точного вираження часу в комунікації.<br>Здатний правильно використовувати часові форми, такі як Present Simple, Present Continuous, Past Simple, Past Continuous, Future Simple, для передачі правильного значення в повідомленні. | **9** | 3 | 1 | 3 | 2 |
| **7.2 Професійна it термінологія в розмові з замовником**<br>Знає і використовує професійні IT-терміни та скорочення, які використовуються в IT-індустрії, для чіткого та зрозумілого спілкування зі замовником.<br>Вміє перекладати складні технічні концепції на доступну мову для замовника, не знайомого з IT-галуззю. | **10** | 3 | 2 | 3 | 2 |
| **7.3 Професійна it термінологія у діловому звіті**<br>Знає і використовує спеціалізовану IT-термінологію та фразеологію при написанні ділових звітів.<br>Вміє передати інформацію про технічні питання та проєкти зрозуміло та конкретно, використовуючи адекватні IT-терміни та абревіатури. | **10** | 3 | 2 | 3 | 2 |

| | | | | | |
|---|---|---|---|---|---|
| **7.4 Професійна термінологія для Data Science**<br>Знає та розуміє специфічну термінологію, що використовується в контексті Data Science<br>Розуміє та використовує специфічні терміни, патерни, алгоритми, фреймворки, бібліотеки та інші інструменти, які використовуються в Data Science | **9** | 3 | 1 | 3 | 2 |
| **7.5 Неформальне спілкування (small talks)**<br>Вміє вести неформальну розмову, розпочинати діалоги та підтримувати невимушену атмосферу.<br>Знає загальновживану термінологію та фрази, які використовуються у неформальних розмовах, таких як загальні питання про погоду, цікаві події, особисті захоплення, спорт, фільми тощо. | **8** | 2 | 1 | 3 | 2 |

## Зауваження від компаній

### Avenga

4.1 Покрити також основні DB engines як для OLAP так і для OLTP
MSSQL, Oracle, PostgreSQL, MySQL
Snowflake, Redshift, Synapse

### GlobalLogic

Загальний коментар від представника Global Logic: всі перелічені навички вище дуже корисні, але очікувати їх від позиції рівня джуніор не варто, це все скоріш мідл-сіньор левел. Для початківця реально важливо просто писати код (ну і початкове розуміння code lifecycle). Тому знань SQL+мова програмування та ще на додаток якась база даних вже буде достатньо, щоб почати щось робити, можливо це не буде максимально ефективно, але вцілом достатньо. Вище проставлено 1 - то більше як nice to have навичка. 0 - на джуніор позиції про це можна навіть не думати.

+ Знання на середньому рівні реляційної бази даних (будь якою з: Postgres, MySQL, MS SQL Server, Oracle). Cloud managed чи unmanaged - неважливо. 3

+ Знання на середньому рівні однієї з мови програмувань - Python, Java, Scala. Як core функціональності, так і специфічних бібліотек: pandas і т.п. 3

+ Знання мови SQL на середньому рівні (в будь-якому контексті: Postgres, Snowflake, dbt, Hive, MySQL, MS SQL Server, Oracle) 3

+ Можна сюди додати зі вкладки Data Science такі секції: 1.1, 1.2, 1.3, 2.1, 2.2, 4.1, 4.2, 4.3, 5.1, 5.2, 5.3

## DataArt

**0. Fundamentals**

**0.1 Дані, інформація, знання**
Розумінян понять дані, інформація, знання (data, information, knowledge) різниці між ними і еволюцію одно в інше — 3

**0.2 Форми даних**
Розуміння понять Structured, semi-structured, non-structured data (здатність навести приклади).
Перехід від одної форми до іншої. Приклади — 3

**0.3 Типи даних**
Знання базових та комлексних типів даних. Розуміння поняття "відсутніх" даних (0 vs NULL vs empty string)
Особливості порівняння різних типів даних. — 3

**0.4 Нормалізація даних**
Знання і розуміння концепції нормалізованих/денормалізованих даних.
**(1NF, 2NF, 3NF).**
Розуміння Primiray Key, Foreign Key. Surrogate Key, Natural Key, Composit Key. Здатність навести приклади — 3

**0.5 Поєднання даних**
Розуміння концепції JOIN та UNION та різниці між ними.
Види JOIN (INNER, OUTER, LIEF, RIGTH, CROSS, ANTI, SEMI) (різниця між ним, приклади, можливі проблеми)
Види UNION — 3(2)

**0.6 Агрегація даних**
Розуміння видів агрегації (повна, часткова/вибіркова, кумулятивна) — 3(2)

**0.7 Якість даних**
Знання і розуміння базових критеріїв якості даних, як то повнота, точність, узгодженість і т.п. — 2

**0.8 Історизм даних**
Розуміння різниці між поточними та історичними даними, та необхідність обох видів в різних ситуаціях.
Знання типів SDC(slowly changing dimension).
Hard delete vs soft delete — 2

**0.9 CDC.**
Розуміння концепції "Changing data capture" та її важливості в різних системах збору/обробки даних — 1

**0.10 Обробка даних**
Знає і розуміє концепції Idempotency, backfulling load, total load vs incremental load vs partial load — 2

**4.6 Архітектура сховища**
Знає і розуміє концепцію багаторівневого сховища даних (e.g. medalion architecture).
Може за наведеним прикладом оцінити і зробити висновки необхідності того чи іншого рівня.
Розуміє проблеми міжрівневих залежностей та складнощі організації пайплайнів з цим пов'язані — 3