```sql
-- Create a staging table similar to the original layoffs table
CREATE TABLE layoffs_staging2 (
  company TEXT,
  location TEXT,
  industry TEXT,
  total_laid_off INT DEFAULT NULL,
  percentage_laid_off TEXT,
  date DATE DEFAULT NULL,
  stage TEXT,
  country TEXT,
  funds_raised_millions INT DEFAULT NULL,
  row_num INT
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

-- Insert data from the original table with row numbers to handle duplicates
INSERT INTO layoffs_staging2
SELECT *,
ROW_NUMBER() OVER (PARTITION BY
    company, industry, total_laid_off, percentage_laid_off, `date`, stage, country, funds_raised_millions)
AS row_num
FROM layoffs_staging;

-- Remove duplicates based on the generated row numbers
DELETE FROM layoffs_staging2
WHERE row_num > 1;

-- Clean up: remove rows where both total_laid_off and percentage_laid_off are NULL
DELETE FROM layoffs_staging2
WHERE total_laid_off IS NULL AND percentage_laid_off IS NULL;

-- Handle missing or blank industry data
UPDATE layoffs_staging2 t1
JOIN layoffs_staging2 t2 ON t1.company = t2.company
SET t1.industry = t2.industry
WHERE (t1.industry IS NULL OR t1.industry = '') AND t2.industry IS NOT NULL;

-- Normalize country data (example for United States)
UPDATE layoffs_staging2
SET country = TRIM(TRAILING '.' FROM country)
WHERE country LIKE 'United States%';

-- Clean up and standardize industry values
UPDATE layoffs_staging2
SET industry = 'Crypto'
```

```sql
WHERE industry LIKE 'crypto%';

-- Convert the 'date' column to a proper DATE format
UPDATE layoffs_staging2
SET `date` = STR_TO_DATE(`date`, '%m/%d/%Y');

-- Query for validation of distinct countries and industries (optional)
SELECT DISTINCT country FROM layoffs_staging2 ORDER BY 1;
SELECT DISTINCT industry FROM layoffs_staging2 ORDER BY 1;

-- Remove unnecessary column row_num after handling duplicates (optional)
ALTER TABLE layoffs_staging2 DROP COLUMN row_num;';


-- Convert date column to proper DATE format
UPDATE layoffs_staging2
SET `date` = STR_TO_DATE(`date`, '%m/%d/%Y');

-- Remove duplicates using row number partition
WITH duplicate_cte AS (
    SELECT *, ROW_NUMBER() OVER (PARTITION BY
        company, industry, total_laid_off, percentage_laid_off, `date`, stage, country, funds_raised_millions)
AS row_num
    FROM layoffs_staging2
)
DELETE FROM layoffs_staging2
WHERE row_num > 1;

-- Standardize industry names
UPDATE layoffs_staging2
SET industry = 'Crypto'
WHERE industry LIKE 'crypto%';
```